The Dissertation Committee for Wei Li

certifies that this is the approved version of the following dissertation:

Data-driven modeling for compound flooding simulation

Committee:

Clint Dawson, Supervisor

Ben Hodges

Irene Gamba

Omar Ghattas

Tan Bui-Thanh

Data-driven modeling for compound flooding simulation

by

Wei Li

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August 2021

Dedicated to my wife, Xinrui Wang.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Clint N. Dawson, for his unremitting guidance, unwavering support, and continuous encouragement over the years of my PhD study. He offered me intellectual, financial, and professional support to broaden my knowledge in computational sciences and to purse the research that interested me. This work benefits from his erudite insights in surface flow modeling. I truly enjoyed working with him while developing my PhD research over the past years.

I want to thank Professors Ben Hodges, Irene Gamba, Omar Ghattas, and Tan Bui-Thanh for serving as my dissertation committee. The discussions with them, reading their work, and taking their classes helped me to develop a broader understanding of environmental fluid physics and computational sciences.

I would also like to thank Dr. Amin Kiaghadi and Dr. Gajanan Choudhary for their guidance and discussions on coupled simulations for compound flooding. I truly appreciate the opportunity to collaborate with them. I wish to thank Dr. Jiachuan He, who has generously and outspokenly shared his experience and wisdom with me through many discussions.

I am thankful for all other members in the Computational Hydraulics Group, my friends and colleagues in the Oden Institute, who made living in and adjusting to a foreign city not so foreign. I am lucky to have each and every one of my friends whose presence has everlasting positive impact on my life.

Last but not least, I am forever indebted to my parents and my wife. Their unconditional love and support have made me who I am. I am glad to have Xinrui Wang as my wife, who has been through a lot of obstacles for us to stay happy and together. Her patience and sacrifices stabilize my life as a PhD student.

Wei Li

The University of Texas at Austin
August 2021

Data-driven modeling for compound flooding simulation

Publication No. _____

Wei Li, Ph.D.
The University of Texas at Austin, 2021

Supervisor: Clint Dawson

Accurate simulation of compound flooding is crucial for flood risk management. In coastal regions, both rainfall runoff and storm surge can contribute to flooding and are not mutually exclusive. It is thus necessary to develop a modeling framework for compound flooding that considers both mechanisms. While many hydrologic models have been developed to simulate the rainfall runoff processes, a lot of these methods are either computationally expensive, or incapable of simulating extreme weather events. Thus, they may not be suitable for coupled modeling of compound flooding. In this study, a data-driven hydrologic model based on deep recurrent neural network (RNN) is developed for rainfall runoff simulation at relatively low computational cost. To test the capability of the method, the model is used to infer the streamflow out of an urban watershed, Brays Bayou in Houston, Texas. And

the model is validated with real world hydrologic data. Additionally, the proposed synced sequence to sequence RNN architecture is compared with the sequence input single output one that is widely-used in hydrologic modeling. Numerical experiments show that the proposed method provides more accurate predictions using relatively less computational resources than the sequence input single output architecture. Later, downstream water level input is integrated into the RNN model to enable the one-way coupling of rainfall runoff with storm surge. Numerical examples at two different locations demonstrate that the additional information leads to improved predictions. Finally, a two-way dynamic coupling framework is constructed for the RNN hydrologic model and an ocean circulation model, ADvanced CIRCulation. The framework is tested, verified, and validated for the Houston ship channel - Galveston bay estuarine system during Hurricane Harvey (2017).

This dissertation is based on the following articles: *High temporal resolution rainfall runoff modelling using Long-Short-Term-Memory (LSTM) networks* by Wei Li, Amin Kiaghadi, Clint Dawson [1]; *Exploring the best sequence LSTM modeling architecture for flood prediction* by Wei Li, Amin Kiaghadi, Clint Dawson [2]; and *Simulating compound floods: dynamic coupling of deep learning and physics-based models* by Wei Li, Gajanan Choudhary, Amin Kiaghadi, Clint Dawson. This material is based upon work funded by National Oceanic and Atmospheric Administration (Grant No. NA18NOS0120158) and National Science Foundation (NSF, CMMI-1520817).

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Compound flooding, that is, flooding due to two or more simultaneous or sequential events, has become more predominant in recent years [3, 4, 5]. The threat from compound flooding is particularly severe in coastal and estuarine regions [6] because of the presence of multiple driving mechanisms such as storm surge, river discharge, and heavy rainfall. Although coastal areas account for less than 10% of the total land in the contiguous United States, it is home to almost 40% of the U.S. population [7]. The impact of compound flooding in these often low-lying, densely populated, and highly developed regions, can be disastrous with extensive social, economic, and environmental consequences.

As an example, on the morning of September 14, 2018, Hurricane Florence, a large and slow moving category one hurricane, made landfall. After the eye crossed Wrightsville Beach, NC at 7:15 a.m. the storm dropped record-breaking rainfall across eastern North Carolina and a portion of northeastern South Carolina for the next two days. The states of North Carolina and South Carolina reported a total of 51 fatalities and estimated damage of $17.3 billion [8]. There is an urgent need for an

accurate modeling framework that effectively integrates all the physical processes involved in such flooding events. In the case of tropical storms, the two major flooding mechanisms would be inland flooding due to intense rainfall and coastal storm surge penetration [9, 10, 11]. Many computational frameworks have been developed to model respective physical processes [12, 13, 14, 15, 16] and to build coupled models that combine them together [11, 10, 17, 18, 19]. However, to the best knowledge of the author, there are few frameworks fully capable of predicting inundation patterns caused by storm surge and rainfall [9, 10].

1.2 Background

The first challenge of compound flooding simulation is developing a reliable and computationally efficient rainfall runoff (RR) model, which produces a surface runoff hydrograph in response to a rainfall event, represented by and input as a hyetograph. Modeling the rainfall runoff process is difficult in itself due to the interactions among multiple physical processes including 1D channel flow, 2D overland flow, infiltration and groundwater flow, precipitation interception, snow melting, and evapotranspiration. Complicating matters further is the lack of software implementation that allows painless coupling of the hydrologic model with storm surge models.

RR models can be categorized as process-driven and data-driven [1, 20, 21, 22]. While process-driven methods are composed of analytical and empirical formulae based on physical phenomena, data-driven models rely on interpolation and extrapolation of recorded data. During the past three decades, multiple process-driven hydrologic models such as Interconnected Channel and Pond Routing Model (ICPR) [15], Hydrologic Engineering Center's River Analysis System (HEC-RAS) [14], and Gridded Surface Subsurface Hydrologic Analysis (GSSHA) [12] for RR simulation have been developed. Although much progress has been made, this class of RR models are restricted by the requirements of accurate geographic data, skilled

users to construct the computational mesh and input files for the model, computationally expensive calibration and inference. To cope with the restrictions of process-driven RR models, a number of recently developed methods have focused on data-driven models. Instead of considering the physical processes during runoff events, data-driven models often map the rainfall hyetograph to the runoff hydrograph in an end-to-end fashion. And the mapping from hyetograph to hydrograph is *learned* from recorded meteorological data. There are many different approaches to construct data-driven RR models. For example, some data-driven RR models are based on multiple linear regression [23], K-nearest neighbor algorithm [22] and support vector regression [24]. Using artificial neural networks (ANN) for real time flood prediction has also been made possible by the advancement of deep learning research. Feedforward neural network has been widely applied to streamflow prediction, e.g. [25, 26, 27, 28, 29, 22, 30].

Due to its capability of modeling highly nonlinear relationships between input and output, the ANN model has generated promising results for RR simulation. When it comes to time series data, standard feed forward neural network has its limitations. Feed forward neural networks are designed based on the assumption that the training samples (data points) are independent. Thus the entire state of the network is erased after processing each data sample [31]. This assumption is not desired when data points are inherently related (with autocorrelation). Moreover, to deal with time series data, a standard ANN model (feed-forward) would require a fixed-sized sliding window over the dataset. Tuning the size of this sliding window for the best predictive accuracy adds extra work to the model selection [25]. This limitation becomes more significant in flood assessment with finer time resolution (e.g. 15 minutes). In this case, long-term dependencies prevail due to the small time step size and cannot be learned by ANN because they are not captured within the fixed-sized time windows.

More recently, a class of ANNs known as recurrent neural network (RNN), a deep learning algorithm, has attracted much attention and shown success in solving sequential problems such as machine translation and speech recognition [32]. Even though the idea of RNN was proposed in the 1980s [33], the applications of RNN in hydrologic engineering are relatively more recent [34, 35]. RNNs are networks with loops in them, allowing information to persist. RNN can exploit the sequential pattern in the data while preserving feed-forward ANN's ability to model nonlinear relationship between input and output via cycles formed by the hidden nodes in the network [36]. A standard RNN has very simple looping units, such as a single layer with hyperbolic tangent (tanh) activation. To cope with the vanishing gradient challenge [37] for standard RNN and learn longer-term dependencies in sequential data, long short-term memory (LSTM) networks based RNN systems have been developed [38, 39]. LSTM's success has encouraged groups to explore its capability in time series forecasting of river discharge. Since 2016, a handful of studies have used LSTM for RR modeling and reported satisfying results [40, 41, 42, 20, 43, 44, 45, 46, 47]. Studies have shown the superior performance of the LSTM network in capturing the dynamics of time-series compared to other RNN networks for hydrologic applications. Table 1.1 provides a summary of those studies and their LSTM architecture.

Reference	Model Type	Resolution	Architecture	Window size
[43]	Time series	Daily	SIO	10
[44]	Regression	Daily	SIO	270
[45]	Time series	Hourly	SIO	1
[42]	Time series	Hourly	SIO	32
[40]	Time series	Daily	SISO	200
[41]	Regression	Daily	SISO	365
[20]	Time series	Monthly	SISO	5
[46]	Time series	Hourly	SIO	8
[47]	Regression	not reported	SISO	not reported

Table 1.1: A summary of studies that used the LSTM network for rainfall–runoff modeling. In architecture colume, SIO stands for sequence input and out, SISO stands for sequence input and sequence output.

However, the existing applications of RNN on RR simulations are either time series forecast, that is using past streamflow to forecast future streamflow, or using a sequence-to-one RNN architecture to predict runoff, in which case the fixed-sized time window is still required (see Table 1.1) as in the feedforward neural network models. In practical hydrologic engineering, sometimes it requires reliable RR simulations for hypothetical events where time series forecast is infeasible due to the lack of past streamflow readings. Moreover, during a rainfall runoff event, gauged streamflow record may not be readily available in real time. Thus, such a model, capable of simulating longer events only using precipitation data as input, is desirable for flood management applications.

The second challenge of compound flooding simulations is to combine all the flooding sources (e.g., local runoff and storm surge) to make reliable predictions. During a compound flooding event, eliminating one of these sources could lead to significant prediction errors. The cascading effects from storm surge and local runoff have not been fully addressed in physics-based storm-surge models such as ADvanced CIRCulation model (ADCIRC) [13] (see e.g. [10, 11]), or RR models such as GSSHA (see e.g. [48]). To cope with this limitation, two or more models need to be coupled together to simulate compound flooding. The main challenge in such a modeling framework is providing the boundary conditions for the storm surge model. While flow gages could be used in gauged watersheds and for the purpose of hindcasting, real-time forecasting could be problematic. Simulating a system with both storm surge and local runoff could be difficult in particular as the model providing the flux (flow) boundary condition could be affected by the presence of storm surge; a dynamic coupling is required. In such a modeling framework, at each time step, the primary model (ADCIRC, for instance) provides feedback to the coupled model (GSSHA, for instance) at the location of flux boundary conditions and vice versa.

In the past decade, numerous efforts on coupling rainfall runoff and storm

surge have been made. In the majority of the existing works, one model is used as the primary model (i.e. recipient model), which takes the results of other model(s) (i.e. courier model) as boundary condition inputs. In this study, this coupling scheme is called the one-way coupling. Among the one-way coupling studies, several used the hydrologic model as the primary model. Silva et al. [17] presented a technique to transfer storm-surge information simulated by the ADCIRC and Simulating WAve Nearshore (SWAN) [49] to the hydrologic model GSSHA. The method was applied to study flooding scenarios occurring during the passage of Hurricane Georges (1998) on the east coast of Puerto Rico. A similar approach was applied in Joyce et al. [50] to explore the worst future case scenario of coastal flooding impacts. The study used ADCIRC/SWAN as the storm-surge model, as well as an integrated hydrological/hydraulic model, ICPR for coastal urban watershed simulation. Another direction of study was to use the ocean circulation model as the primary model, with ADCIRC being the most commonly used storm surge model. In this case, freshwater discharges have been used as input for the ocean circulation model, in addition to the wind field and atmospheric pressure data, by means of boundary conditions. The discharge information either comes from gage readings [18, 51, 52], or a hydrologic model [53].

On the other hand, the technique that allows transfer of information occurring in both directions is called the two-way coupling scheme. In the loosely two-way coupling case, the result of one model (e.g. river discharge and seawater levels) are used as boundary conditions for the other model. To the best knowledge of the authors, only three works [54, 19, 11] studied the two-way coupling of both mechanisms. Cheng et al. [54] coupled an ocean circulation model (i.e. ADCIRC) with a hydrologic model (i.e. pWASH123D) using boundary condition points. The impact of Hurricane Katrina (2005) over the Mississippi coast was simulated using a synthetic rainfall instead of the actual hurricane rainfall. This method assumes

the interaction between the two models occurs at the coastline only. However, this interaction could naturally occur upstream in the river outlet or in the coastal floodplain. Tang et al. [19] coupled a hydrologic model (i.e. Flood Potential Model) and hydraulic/ocean circulation model (i.e. FVCOM and Shallow Water Model) in a tighter fashion. This approach is more complicated than the boundary condition exchange since it involves implementing and modifying existing code of numerical models. The complexity limited the authors to not consider any infiltration, precipitation losses, surface routing scheme or runoff volume computations which in some cases contributes to improved results.

More recently, several efforts have been made to apply machine learning techniques in compound flooding. Benjamin & Bedient [55] developed surrogate models based on ANNs and the Kriging regression for a physics-based loosely-coupled compound flooding model. The surrogate models are trained with data generated by the physics-based model and are considered as a more efficient substitute for it. Park & Li [56] developed a coastal flooding risk prediction model with k-nearest neighbors algorithm. The model predicts the probability of compound flooding hazard for several specific locations. French et al. [57] developed an ANN that generates short-term forecasts of water levels at estuarine ports based on distant tide gauge observations, wind and atmospheric pressure, and the predicted astronomical tide. The forecast is then used as boundary condition for a local high-resolution hydrodynamic model that predicts flood extent and estimates damage potential. However, none of the aforementioned studies have truly dynamically coupled the machine learning and physical models in a two-way fashion. In other words, none of these studies have feedback from the machine learning and physics models being exchanged at each simulation time step.

In this work, a modeling framework for compound flood simulation that couples a data-driven rainfall runoff model and the process-driven ocean circula-

tion model (i.e. ADCIRC), is considered. Since the required RR model is of high temporal resolution, implementing it with RNN serves as a robust test of the ability of RNN to preserve long-term memory for regression tasks [1]. Furthermore, among the different RNN architectures, this study demonstrates that one particular class of RNN architectures is more attractive for compound flood modeling [2]. The RNN architecture can then be used to develop a one-way coupled RR model that accounts for local runoff and downstream tides and storm surge. Eventually the coupled RR model can be integrated into a two-way dynamically coupled compound flooding model. The coupling framework that combines the data-driven RR model and ADCIRC is based on a dynamic coupler for computational hydrologic and hydrodynamic models [11] with using Python interface of ADCIRC [58].

The rest of the dissertation is organized as follows. In Chapter 2, a brief overview of RNN and LSTM network is followed by the introduction of a data-driven RR model using RNN. Its performance is compared with a process-driven RR model (i.e. GSSHA)[1]. The RNN architecture used in the developed data-driven hydrologic model is then compared with another RNN architecture, which is widely used in hydrologic modeling, in Chapter 3[2]. Chapter 4 presents one-way coupled compound flooding model based on the aforementioned RR model followed by its integration into a two-way dynamic coupling framework[3]. Finally, some concluding remarks are provided in Chapter 5.

[1]Chapter 2 is based on the article entitiled *High temporal resolution rainfall–runoff modeling using long-short-term-memory (lstm) networks* by Wei Li, Amin Kiaghadi,and Clint Dawson [1]

[2]Chapter 3 is based on the article entitled *Exploring the best sequence LSTM modeling architecture for flood prediction* by Wei Li, Amin Kiaghadi,and Clint Dawson [2]

[3]Chapter 4 is based on the article entitled *Simulating compound floods: dynamic coupling of deep learning and physics-based models* by Wei Li, Gajanan Choudhary, Amin Kiaghadi,and Clint Dawson (submitted to Computational Geosciences)

Chapter 2

RNN hydrologic model

Traditionally, rainfall runoff models are process-driven; i.e. they solve either simplified empirical formulas or some variant of the St. Venant (shallow water) equations numerically. Such equations are often parameterized based on geographic data (e.g. surface roughness based on land use data). To construct a process-driven model, nontrivial engineering effort and expertise are required to collect up-to-date geographic data and construct a computational mesh accordingly before the numerical equations can be solved. Moreover, such models requires extensive calibration of parameters such as the surface roughness coefficient. Solving the equations numerically can be computationally expensive and calibrating the parameterized equations, which requires solving the forward equations many times, poses a greater challenge. With the development of deep learning techniques, it is possible to emulate rainfall runoff models using, for example, deep recurrent neural networks. Since, at the inference time, the RNN consumes negligible computational resources, it is particularly suitable for coupled compound flooding simulation when the only required output

This chapter is based on published article Li, W., Kiaghadi, A. & Dawson, C. High temporal resolution rainfall–runoff modeling using long-short-term-memory (LSTM) networks. Neural Comput & Applic 33, 1261–1278 (2021). https://doi.org/10.1007/s00521-020-05010-6. The author of this dissertation proposed using synced sequence to sequence RNN to model rainfall runoff, implemented the model, and conducted the numerical tests.

is the hydrograph of the watershed. In this chapter, a data-driven RR model using a Long short-term memory (LSTM) network is presented. And its performance is evaluated and compared with a process-driven model GSSHA.

2.1 Study area and datasets

Figure 2.1 shows the location of Brays Bayou watershed and its tributaries located in southwest Harris County and northeast Fort Bend County, Texas selected for this study. Brays Bayou drains freshwater from 329 square kilometers of a heavily urbanized and populated watershed and discharges into the Houston Ship Channel [59]. Brays Bayou has had a history of floods; just in the last 18 years Tropical Storm Allison (2001), Hurricane Ike (2008), the Memorial Day Flood (2015), the Tax Day Flood (2016), and Hurricane Harvey (2017) caused significant flooding and billions of dollars of property damage [60].

15-minute precipitation data from 2007 to 2017 were compiled from 153 rainfall gauges maintained by the Harris County Flood Control District (HCFCD) and 15 minutes flow data were obtained from the United States Geological Survey (USGS) gages [61]. Within the Brays Bayou watershed there are 15 rainfall gauges and five flow gages. In this study only one freshwater gage located very close to the watershed outlet (see gage 08075000 in Figure 2.1) was used to compile flow data for the purpose of training, validation, and evaluation. To handle the missing data, a threshold of 90-minutes was set. If the missing data gap was less than or equal to 90 minutes (6 missing points), the missing points were imputed by linear interpolation. For gaps greater than 90 minutes, the sequence was split at the gaps. To speed up training and avoid the gradient exploding problem [62], the time series is further split into even shorter series. In order to eliminate the scale difference among the input features, a minimum-maximum scaling was applied to both input and output variables [25, 28].

To construct the GSSHA model for the Brays Bayou watershed, land elevation was extracted from the 10m resolution U.S. National Elevation Dataset (NED) in Watershed Modeling System (WMS) and assigned to the grid. The 15-class land use data was compiled from 30m resolution U.S. National Land Cover Database (NLCD).

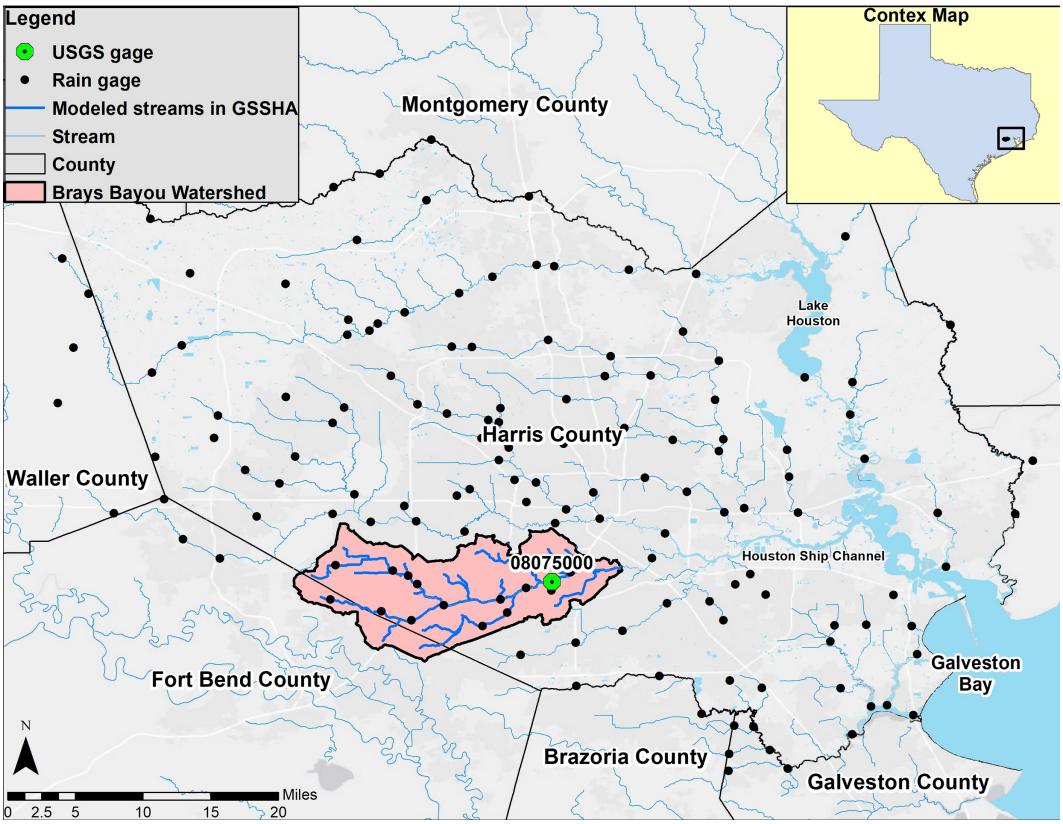


Figure 2.1: Study Area: Brays Bayou in Harris County, Texas.

2.2 Deep recurrent neural network and long short-term memory network

Recurrent neural networks (Figure 2.2), a class of ANNs with loops in them, are designed for tasks that involve sequential inputs, such as speech and language [63]. At each time step t, the neural network A looks at some input $X_t \in \mathbb{R}^d$, where d is the dimension of the input, and hidden state from the last time step h_{t-1}, and outputs a new hidden state h_t. A loop allows information to be passed from one step of the network to the next. At the next time step $t+1$, the new input X_{t+1} and hidden state h_t enters the network, and new hidden state h_{t+1} is computed. In this way, a RNN can map an input sequence with element X_t into an output sequence with elements h_t, with each h_t depending on all the previous $X_{t'}$ ($\forall t' < t$). And the same parameters of A are used at each time step.

Training an RNN is not fundamentally different from training simple feedforward ANNs. Using backpropagation through time [64], which is nothing more than a practical application of the chain rule for derivatives, the gradient of an objective function with respect to the weights of the RNN can be obtained. And gradient based optimization algorithm will be applied to the objective function to find the optimal set of parameters.

In theory, RNNs are capable of handling "long-term dependencies". For instance, initial input X_0 could affect the hidden state value 500 steps later (h_{500}). Although the RNNs seem to be powerful and straightforward, training them can be problematic when more time steps are included in a sequence. As the backpropagated gradients either grow or shrink at each time step, over many time steps the growth or shrinkage typically leads to gradient explosion or vanishing [37, 65]. Depending on the size of the watershed, the peak of generated runoff can be observed from a couple of hours to a couple of weeks after the event. For instance, for a time step of 15 minutes, a rainfall event that lasted for a week would have more

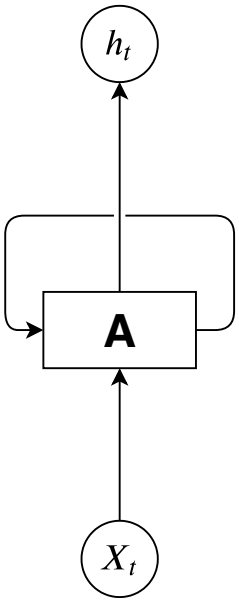


Figure 2.2: Illustration of recurrent neural network.

than 650 steps in time. To accurately model the RR process, the model is required to memorize the effect of precipitation from the beginning of the event, which is numerically difficult for standard RNN. Thanks to advances in their architecture [39, 66] and ways of training them [67, 68], training RNNs with longer sequences became feasible.

Among the variants of RNNs, LSTM networks [39] have been one of the most widely used due to their capability to learn long-range dependencies. Suppose we are given a set of l training sequence samples, $\{(X_1, y_1), ..., (X_l, y_l)\}$, where $X_i \in \mathbb{R}^{m \times n}$ is an input sequence with m steps and n elements in each step, $y_i \in \mathbb{R}^m$ is the target output sequence. As a first step, a LSTM unit maps the inputs at time t (i.e. $X_t \in \mathbb{R}^n$, previous hidden state h_{t-1}, and previous cell state c_{t-1}) to a set of

components including the cell state $c_t \in \mathbb{R}^n$, the input gate $i_t \in \mathbb{R}^n$, the cell gate $g_t \in \mathbb{R}^n$, and the output gate $o_t \in \mathbb{R}^n$. The mappings take the form:

$$\begin{aligned} i_t &= \sigma(W_{ii}X_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\ f_t &= \sigma(W_{if}X_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\ g_t &= \sigma(W_{ig}X_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \\ o_t &= \sigma(W_{io}X_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\ c_t &= f_t * c_{t-1} + i_t * g_t \end{aligned} \tag{2.1}$$

where $\sigma(\cdot)$ is the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$, W_i's are the input-hidden weight matrices, and W_h's are the hidden-hidden weight matrices, the b_i's are the input-hidden bias vectors, and b_h's are the hidden-hidden bias vectors, and $*$ is the Hadamard product. The new hidden state h_t is then computed based on output gate and cell state:

$$h_t = o_t * \tanh(c_t) \tag{2.2}$$

Note that there can be multiple layers of LSTM network (see Figure 2.3), also known as deep RNN/LSTM. In this case, the previous layer's hidden state h_t^{s-1} would be the input to the current layer, and the mapping from h_t^{s-1}, h_{t-1}^s, and c_{t-1}^s to h_t^s and c_t^s is the same as defined in equation 2.1 and 2.2. The output of the forward propagation of the network can be written as $g(h_t)$: $\hat{y}_t = g(h_t)$. There are multiple approaches to define $g(h_t)$ depending on different applications, for example, a linear layer with activation function f_a (see Figure 2.3):

$$g(h_t) = f_a(b + Wh_t) \tag{2.3}$$

In the context of artificial neural networks, the activation function of a node defines the output of that node for a given input or set of inputs. A nonlinear activation

function allows the neural networks to model nonlinear relationship between the input and output. This function is also known as the transfer function. A widely used activation function that is defined as the positive part of its argument, is called rectifier [69] or rectified linear unit (ReLU):

$$f_a(x) = x^+ = max(0, x), \tag{2.4}$$

For the sake of simplicity, we denote the map from X_t to y_t defined by equations 2.1-2.3 as $y_t = f(X_t)$. Finally, an objective function $\mathcal{L}(\hat{y}; y)$ (also called loss function) can be defined, for example, the mean squared error (MSE):

$$\mathcal{L}(X, y; \mathbf{w}) = \frac{1}{2}||f(X) - y||_2^2 \tag{2.5}$$

The regression problem can then be mathematically formalized as minimization of the objective function w.r.t the parameters of the neural network $\mathbf{w}$.

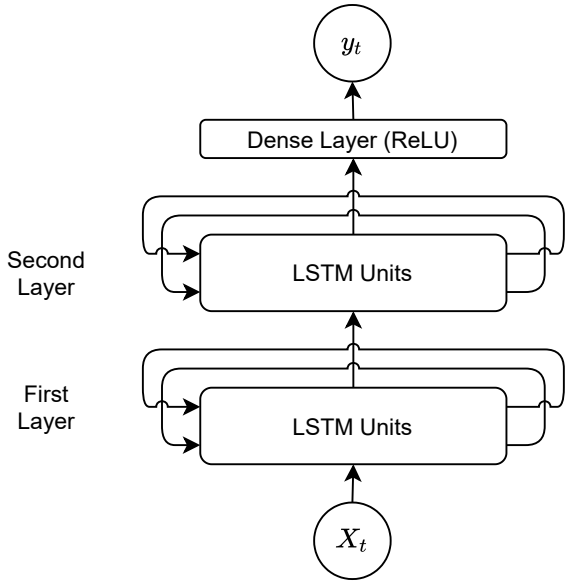


Figure 2.3: Developed LSTM network.

To solve the above optimization problem, most practitioners use a procedure called stochastic gradient descent (SGD). SGD is an iterative method for optimizing an objective function with suitable smoothness properties (e.g. differentiable or subdifferentiable) [70]. It can be regarded as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient by an estimate thereof (calculated from a randomly sampled subset of the data). Adaptive gradient methods such as adagrad have gained widespread use in large-scale optimization for their ability to converge robustly, without the need to fine-tune the stepsize schedule [71]. Among the variants of the adaptive gradient methods, Adam [72], first published in 2014, has become one of the most popular first order optimization algorithms and additional research [73, 74] has been done to further improve the original version of the algorithm.

Regularization, a process in which information is added in order to solve an ill-posed problem or to prevent overfitting, is crucial in building machine learning models [75]. Since the recurrent neural network is extremely flexible, appropriate regularization techniques are necessary to constraint the variance of the trained model. One of the most popular regularization method is adding l_2 penalty to the objective function, namely, instead of the loss function itself, the objective function becomes:

$$\min \mathcal{L}(X, y; \mathbf{w}) + \mathcal{R}(\mathbf{w}) \tag{2.6}$$

where $\mathcal{R}$ is defined as:

$$\mathcal{R}(\mathbf{w}) = \frac{\lambda}{2}||\mathbf{w}||_2^2 \tag{2.7}$$

and λ is the regularization parameter. Larger λ corresponds to more regularization. Note that this technique is also known as weight decay because when applying standard SGD, it is equivalent to updating the weight in this way:

$$w_{i+1} = w_i - \lambda w_i - \alpha \frac{\delta L}{\delta w}|_{w_i} \tag{2.8}$$

where w_i is the learnable parameters at step i, α is the learning rate, and $\frac{\delta L}{\delta w}|_{w_i}$ is the stochastic gradient approximation at step i. Thus, at each step, the weight w *decays* by $(1 - \lambda)$. However, l_2 regularization is not equivalent to weight decay in the adaptive methods including Adam optimization; state-of-the-art deep learning frameworks such as PyTorch [76] adopts the implementation of l_2 regularization proposed in [73]. Another widely used regularization for ANNs is dropout [77]. In a multilayer LSTM, the input X_t^l of the l-th layer ($l \geq 2$) is the hidden state h_t^{l-1} of the previous layer multiplied by dropout δ_t^{l-1}, where each δ_t^{l-1} is a Bernoulli random variable which is 0 with a user specified probability p and 1 with probability $1 - p$. The core concept of applying dropout is to force each hidden unit in a neural network to learn to work with a randomly chosen sample of other units. Srivastava et al. [77] hypothesize that by making the presence of other hidden units unreliable, dropout prevents co-adaptation of each hidden unit.

2.3 Training, validation, and evaluation of RNNs for runoff models

In machine learning, a mathematical model is constructed from existing data. However, the task of the machine learning model is to make predictions on future data that is not available at the model construction time. To evaluate the model performance on unseen data, a common practice in supervised machine learning is to split the data into three datasets which are used in different stages of the creation of the model.

Specifically, the model is initially fit on a training dataset, that is a set of data samples used to fit the parameters (i.e. $\mathbf{w}$ in equation 2.6) of the model. Successively, the fitted model is used to predict responses for the observations in a second dataset called the validation dataset (e.g. predict hydrograph given precipitation in this

study). The validation dataset provides an unbiased evaluation of the model fit on the training dataset while tuning the model's hyperparameters (e.g. the number of the hidden units in neural network, number of LSTM layers, regularization, type of activation function, etc.). The combination of the hyperparameters with the best validation performance is then chosen for the machine learning model. Finally, the test dataset is used to provide an unbiased evaluation of a final model. If the examples from the test dataset have never been revealed to the model during training and validation stages, the test dataset is also called a holdout dataset.

In this study, hydrologic data was split into train, validation, and test datasets. As shown in Figure 2.4 all 15-minute data up to the end of 2015 (2007-2015) was used for training. The entire year 2016 was used for validation and 2017 was used as the holdout test dataset. This train-validation-test split scheme is designed to minimize over-fitting and consistent with realistic prediction scenarios.

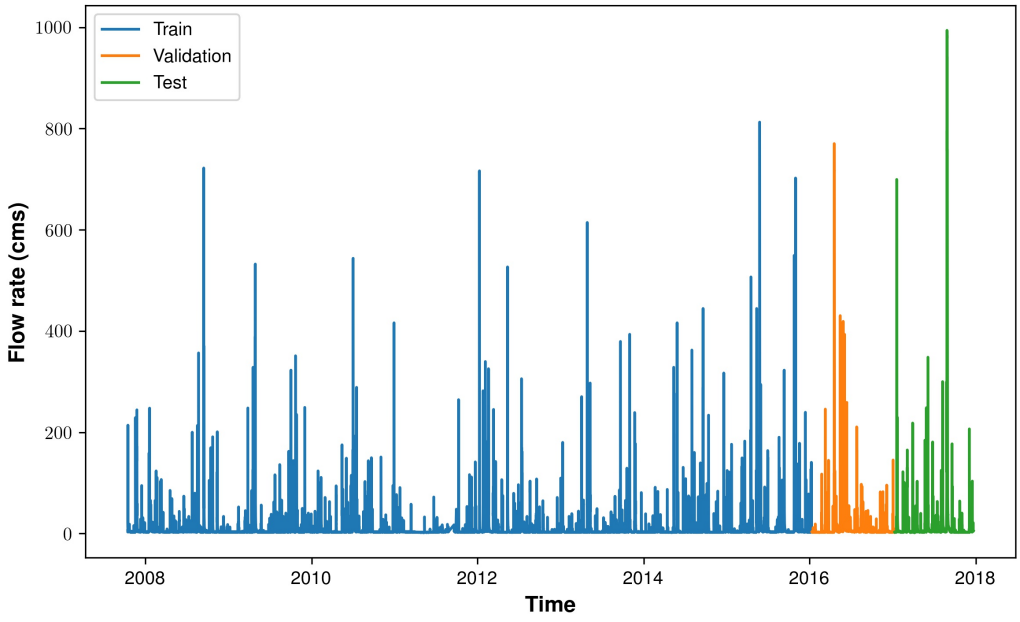


Figure 2.4: Illustration of training, validation, and test data split scheme.

At the training stage, an training *iteration* includes a forward propagation of the training example that computes the output, a backward propagation that computes the gradient, and an optimization step that updates the learned parameters $\mathbf{w}$. Multiple training examples can be put into a *batch* where both forward and backward propagation are processed in parallel respectively. One forward pass and one backward pass of all the training examples is called an *epoch*.

During training, each sequence of data from the training dataset was input into the LSTM network as one batch. Built-in Adam optimization algorithm was used to optimize the MSE loss function. To avoid over-fitting, an l_2 regularization was added for all learnable parameters.

After the RNN model is *trained*, it is evaluated with data from validation dataset for hyperparameter tuning. Since this study focuses on RR prediction for flood events, the evaluation was focused on flood events instead of normal flow regime dominated by the base flow and tidal mechanisms. The Nash–Sutcliffe model efficiency (NSE) and root-mean-square error (RMSE) were selected as the metrics for model evaluation. Note that the developed RNN models (as well as GSSHA model) is not time series forecasting models. Hence time series specific metrics such as persistency criterion which compares the forecasting with the predictions from a naive persistence model are irrelevant and not used in this study. Using a manual search approach, a two-layer LSTM network with 10 hidden units in each LSTM layer was selected (see figure 2.3). The best learning rate was 10^{-4} (α of the Adam optimizer) and the best l_2 regularization was 10^{-6}. Finally, the performance of the selected model can be evaluated on the test dataset.

2.4 GSSHA model for Brays Bayou

GSSHA is developed and actively operated by the Engineer Research and Development Center (ERDC) of the United States Army Corps of Engineers (USACE).

GSSHA is an open source distributed-parameter hydrologic model capable of coupling multiple physical interactions among 1D channel flow, 2D overland flow, infiltration and groundwater flow, precipitation interception, snow melting, and evapotranspiration [78]. Among the process-driven models, GSSHA has been widely used by many researchers for various purposes from total maximum daily loads (TMDLs) to compound flooding; in the period of 2000-2017, GSSHA has been used in more than 85 scientific/technical projects [79]. More recently, [17] coupled GSSHA with the state-of-the-art surge modeling system (ADCIRC-SWAN) to simulate compound flooding on the east coast of Puerto Rico. Other recent studies used GSSHA to improve the parameterization of the Storm Water Management Model [80], and to evaluate the performance of satellite-based precipitation products in comparison to radar data [81]. Thus it is chosen as the benchmark process-driven model for this study. Considering the location, geography, and objectives of the study only surface flow routing processes were activated.

In this study, a GSSHA model was built for the study area using the WMS version 10.1. WMS is a watershed RR simulation and modeling software application from Aquaveo™[82]. The software supports a number of hydraulic and hydrologic models including GSSHA that can be used to create drainage basin simulations. A uniform 2-D grid with 56,606 cells with a dimension of 100 by 100 meters for 2D overland flow was constructed. The streams were represented by 49 reaches of trapezoidal channels. The channel nodes have an average length of 470 meters in the longitudinal direction. The cross-section geometry is approximated based on an existing HEC-RAS model for Brays Bayou developed by the USACE. To compute 2D overland flow, the alternating direction explicit (ADE) method was chosen in GSSHA. To assign surface roughness parameters (Manning coefficient) an index map was created using 15-class land use data. For each land use class (see Figure 2.5 in the supplementary information (SI)), Manning coefficient recommended by

the National Resource Conservation Service (NRCS) was used (see Table 2.1). The channel flow is modeled using explicit diffusive wave method. The precipitation data was obtained from 15 rainfall gauges inside the Brays Bayou watershed. The distributed rainfall was then interpolated using Theissen polygon and inverse distance weighted methods.

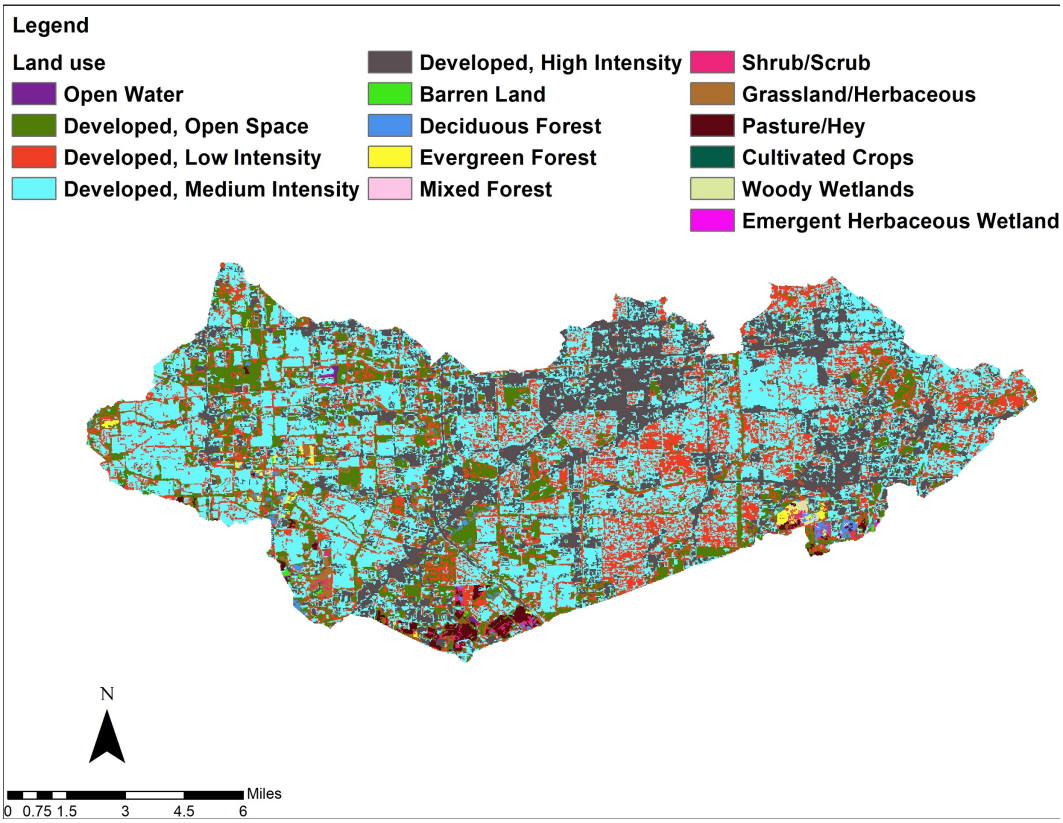


Figure 2.5: Land use of study area.

Limited by the computational expense, the GSSHA model used in this study was calibrated on an event from 11/17/2016 to 11/27/2016. The peak flow rate of this event was 82.69 m^3/s, which was close to that of the moderate rainfall event. River channel's Manning's coefficient was set as the only calibration factor and restricted within the range between 0.001 and 0.02. The GSSHA built-in automated

NLCD Value	Normal Manning's n value	Allowable range of n values
11	0.040	0.025-0.05
21	0.040	0.03-0.05
22	0.100	0.08-0.12
23	0.080	0.06-0.14
24	0.150	0.12-0.20
31	0.025	0.023-0.030
41	0.160	0.10-0.16
42	0.160	0.10-0.16
43	0.160	0.10-0.16
52	0.100	0.07-0.16
71	0.035	0.025-0.050
81	0.030	0.025-0.050
82	0.035	0.025-0.050
90	0.120	0.045-0.15
95	0.070	0.05-0.085

Table 2.1: Manning's n values for various land covers to use for GSSHA simulation.

calibration tool using Levenberg Marquardt (LM) / Secant LM (SLM) was chosen as the optimization algorithm for model calibration. Optimization started with Manning's n at 0.02 and root mean squared error (RMSE) at 7.90 m^3/s. After 12 model runs, 0.003 was found to be the optimal Manning's coefficient. The RMSE of the final model's prediction was 6.45 m^3/s.

Furthermore, four flood events in 2017 with different scales were simulated using the calibrated model to be compared with the LSTM model's result and observed data. To be consistent with the data-driven model, discharge at the chosen USGS fresh water gage (08075000) is set as the observation point.

2.5 Numerical Experiments

2.5.1 Statistical correlation and physics intuition

Physically, it is obvious that the amount and pattern of rainfall collected by gauges near/upstream of the flow gage are more relevant to runoff discharge than those

collected by gauges that are far from/downstream of the river gage. To explore how the LSTM model results are spatially distributed and verify that the trained data-driven model is consistent with the physical intuition, two numerical tests were conducted using precipitation data from all 153 aforementioned rainfall gauges:

1. Precipitation data from each rainfall gauge was used to train a separate LSTM model. Then the training loss of all of the 153 models were recorded and compared. The assumption was: if the LSTM model could actually learn the physical correlation between precipitation and river discharge, the model trained with more relevant input data should perform better.

2. All gages' data were used to train a single LSTM model. It is natural to assume that a physically consistent model should pay more attention to the more *important* gages.

The purpose of these two numerical tests is to explore the characteristics of the LSTM network for the defined application and reduce the number of input gauges to the model based on the physical intuition of the problem.

For the first test, each model was trained for 200 epochs on the training dataset combined with the validation dataset and the best performing epoch with the minimum training error was recorded. Because the models trained in this test was not intended to be used for prediction, regularization and validation were not applied in this case. For the second test, due to the higher input dimension more epochs were required for the model to converge. Thus, the all-gauge model was trained for 400 epochs on the training dataset. The best performing epoch on the validation dataset was chosen as the trained model. Regularization and validation were applied here to (1) cope with the ill-conditioning problem when highly correlated precipitation data from different gages are presented; (2) prevent over-fitting so that the model parameters, including the first layer weights, come from a meaningful model. As noted before, the first layer of the LSTM network takes precipitation input. After

the training, all the learnable input-hidden weights (W_{ii}, W_{if}, W_{ig}, and W_{io}) of the first LSTM layer are grouped by gauges and then flattened to a vector W, i.e.

$$W = \{W_{ii,1}, W_{if,1}, W_{ig,1}, W_{io,1}, ..., W_{ii,n}, W_{if,n}, W_{ig,n}, W_{io,n}\} \tag{2.9}$$

where n is the number of hidden units, i.e. 10. For each gauge, three parameters of the learnable input-hidden weights were defined by the l_1, l_2, and l_∞ norms of W:

$$||W||_1 = \sum_{i=0}^{n}(|W_{ii}| + |W_{if}| + |W_{ig}| + |W_{io}|) \tag{2.10}$$

$$||W||_2 = [\sum_{i=0}^{n}(W_{ii}^2 + W_{if}^2 + W_{ig}^2 + W_{io}^2)]^{1/2} \tag{2.11}$$

$$\begin{aligned} ||W||_\infty = \max\{&|W_{ii,1}|, |W_{if,1}|, |W_{ig,1}|, |W_{io,1}|, ..., \\ &|W_{ii,n}|, |W_{if,n}|, |W_{ig,n}|, |W_{io,n}|\} \end{aligned} \tag{2.12}$$

Thus, for each gauge, this test generates four parameters: the training error e, and the three norms of the weight vector W. To find any correlation, if any, among the three norms of the weight vector and training errors from the first numerical test, a correlation analysis was conducted using both Pearson correlation coefficient (r) and Spearman's rank correlation coefficient (ρ).

The training errors of LSTM models using each single rainfall gauge (first numerical test in section 2.5.1) are shown in Figure 2.6. The lowest training error was 29.94 in a gauge just upstream of the discharge gage and highest training error was 278.11 in a gauge located outside of Harris County. From Figure 2.6 it can be seen that gages with the best performance are the ones located within or near the watershed. In fact, the Pearson correlation between the training error extracted from the LSTM model and the physical distance between the rainfall gauges and the USGS gage was statistically significant with a p-value of 3.5E-31 and r=0.77. This

results show that similar to the process-driven model, where precipitation drives runoff and the amount of precipitation falls into the watershed is represented by the interpolation of the rainfall gauge recording, the data-driven model also performs better when better representation of the distributed precipitation is provided.

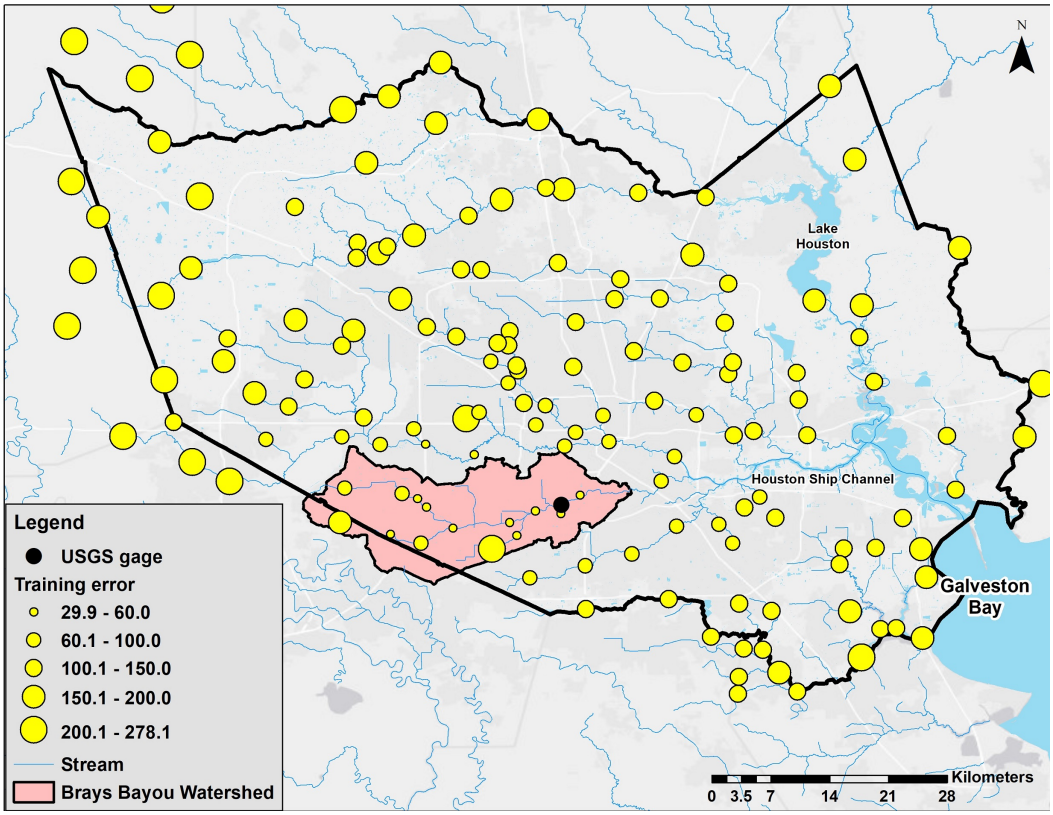


Figure 2.6: Training error map for single gage models.

The correlation between training error e and the three parameters defined in equations 2.9 - 2.12 can provide an intuition on how much attention the LSTM model pays to the important rainfall gauges. This makes more sense when considering the fact that the spatial distribution of the *best performing gauges* matches very well with the physical intuition. The Pearson correlation coefficient, Spearman's rank correlation coefficient, and the respective p-values of the statistical t-tests are shown

in Table 2.2. All statistical t-tests suggest it is safe to reject the null hypothesis that the weight parameter is uncorrelated with the gauge training error. Moreover, there is a non-trivial negative correlation between the norms of first layer weights and performance of model trained using the corresponding gauge. The result suggests that statistically the LSTM model pays more attention to the *physically important* gauges than those irrelevant gauges.

Table 2.2: Correlation between gage training error and weights parameters.

Parameter	r	p-value of r	ρ	p-value of ρ
$\|\|W\|\|_1$	-0.731	7.551e-27	-0.335	2.264e-5
$\|\|W\|\|_2$	-0.742	4.457e-28	-0.343	1.389e-5
$\|\|W\|\|_\infty$	-0.664	9.071e-21	-0.318	6.097e-5

As suggested in [83, 84], eliminating redundant gauges effectively improves the predictions of RR models. The consistency not only suggests the LSTM model is paying more attention to the more important gauges, but also provides an efficient way of choosing rainfall gauges for the LSTM model. Unlike the dedicated studies of choosing rainfall gauges for RR modeling using areal rainfall optimization [84], the LSTM model can provide a coarse yet fast approach to pick the most relevant gauges.

2.5.2 Feature selection

To reduce the training time and the need for input data, it is necessary to reduce the number of rainfall gauges used for the training. In addition, reducing the number of gauges should not impair, if not improve, the performance of the model. An exhaustive feature selection would require trying all combinations of gauges which means training 2^{153} models which is infeasible. Thus, the choice of rainfall gauge was the 10 most relevant gauges from section 2.5.1 using the gauges with the minimum training errors.

A slightly different training process for this test was followed since this model was expected to be used for prediction. The LSTM model was trained on the training dataset and regularization was added. The number of epochs were restricted to 200 and the best performing (in terms of evaluation score) epoch on the validation dataset was chosen as the training result.

To show the causal improvement of this feature selection approach, more numerical tests were conducted. Comparison was made among models trained with the 10 best gauges (based on training error), 10 randomly sampled gauges (sampled 5 times) from all 153 gauges, 10 randomly sampled gauges within the watershed, and 10 closest gauges to the discharge gage. Thus, a total of eight models using 10 rainfall gauges were trained and tested.

Using the feature selection criterion suggested in section 2.5.1, the 10 gauges with the lowest training error in the first test, which are mostly located inside the watershed (the only gauge outside the watershed is also very close to the watershed boundary), were picked as the input of the finalized LSTM model. To show the causal improvement of feature selection, the model performance is also compared (see Figure 2.7) with models using (1) 10 randomly chosen gauges within Harris County (sampled 5 times) (2) 10 randomly chosen gauges within the watershed (3) closest 10 gauges. The result in Figure 2.7 shows that the best 10 gauges model not only has the best validation score, but also converged faster than all other aforementioned models. Note that the model with 10 randomly chosen gauges within the watershed also has a high validation score (0.945) but this model shares 6 common gauges with the best 10 gauges model. Nonetheless, its best epoch is 296 which is more than twice as much as that of the best 10 gauges model indicating longer computational time for training.

Compared to the 153-gauge model in experiment 1, training of the best 10 gauges model converged significantly faster: it took 145 epochs to converge whereas

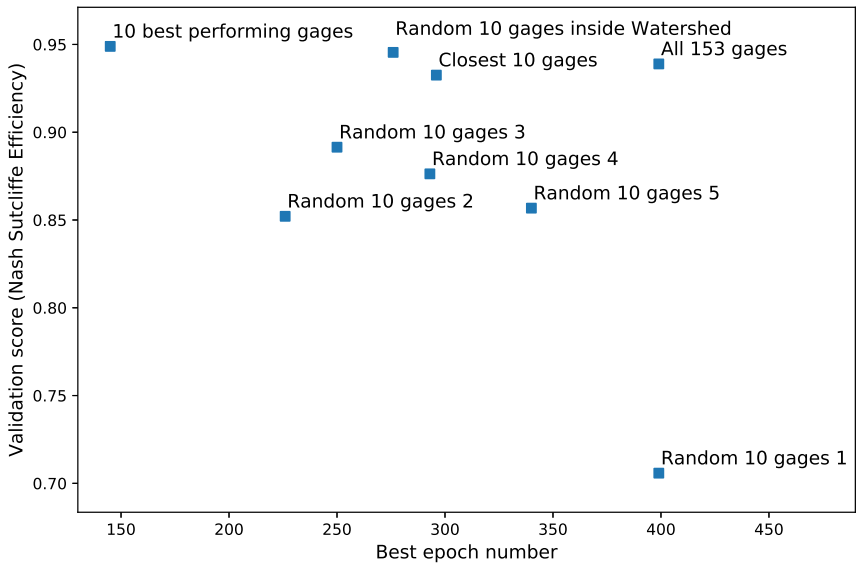


Figure 2.7: Validation score vs number of epochs.

the 153-gauge model required more than 350 epochs to converge using the same learning rate. The convergence analysis of Adam optimization algorithm is out of the scope of this dissertation, however, it is clear that the convergence of Adam algorithms is dimension dependent [72]. Besides, lower dimension implies lower computational cost for each iteration. Hence, training models with fewer inputs would be more efficient.

The evaluation scores (RMSE and NSE) were computed for both 10-gauges model and 153-gauges model on training/validation/test dataset and are shown in Table 2.3. The 153-gauges model had lower training error but higher validation/test error compared to the 10 gauges model which implies more over-fitting of the 153-gauges model.

Moreover, the test error of both models were significantly higher than the training and validation errors. However, this behavior is explainable and does not

Table 2.3: Evaluation scores of 10-gauges model versus 153-gauges model.

dataset	10-gauges model		153-gauges model	
	RMSE	NSE	RMSE	NSE
Training	8.11	0.921	5.65	0.961
Validation	7.83	0.947	8.49	0.938
Test	17.62	0.942	42.24	0.666
Test excluding Hurricane Harvey	8.32	0.906	10.04	0.864

indicate our model is over-fitting. The larger test error was dominated by under-predicting Hurricane Harvey which was included in the test set (see Figure 2.8c). It should be noted that, Hurricane Harvey was an extraordinary flooding event in which, due to the high volume of precipitation, inter-basin transfer happened in many of the watersheds in the Greater Houston Area. Such phenomena is almost impossible to capture even with process-based models when only one watershed is modeled. Considering the uniqueness and rarity of Hurricane Harvey and its different behavior in both precipitation pattern and volume, the prediction of the 10-gauges model on Hurricane Harvey, as shown in Figure 2.9d, is acceptable. The test result actually shows the relatively good extrapolation ability of the data-driven model.

From Figure 2.8, it can be seen that the test set contains target discharge above 900 m^3/s, while the training/validation sets have lower peak flow rate. The 10-gauges model clearly performs better than the 153-gauges model on an extreme event (Harvey). The flow rate versus time plots of both models for every event in the test dataset with peak flow larger than 30 m^3/s are shown in the Figure 2.10. It can be seen that for the majority of the events except Harvey, both models are making reasonable predictions. Table 2.3 shows that if the time series containing Hurricane Harvey was excluded from the test set, the performance of the 10-gauges model would be closer to those on the training and validation set. However, compared to the 10-gauges model, the 153-gauges model still seems to have larger variance

given that it has better training score (compared to 10-gauges model) but worse validation/test score.

2.5.3 Comparison of RNN and GSSHA models

To compare the LSTM model with the benchmark GSSHA model, four events of different scales in terms of precipitation and river discharge were chosen from the test dataset (2017); low rainfall event from 9/28/2017 to 10/6/2017, moderate rainfall event from 12/16/2017 to 12/19/2017, high rainfall event from 12/2/2017 to 12/12/2017, and finally, an extreme rainfall event including Hurricane Harvey, which started from August 23 and ended on September 1, 2017. Note that the moderate event follows the high event with an interval of 4 days. Here it was assumed the precipitation of the first event had completed runoff by the time the second event starts. The choice of these four events were due to the limitation of computational expense of the GSSHA model, not the LSTM model. The LSTM model has no such restriction and was tested for all of 2017's flood events.

Thanks to the superior performance of the 10-gauge LSTM model chosen through feature selection, this model was used as the final data-driven model in this study. Thus, for the rest of this chapter, the 10-gauge model is referred as the LSTM model unless indicated otherwise. Figure 2.9 shows the comparison between the predicted flow rates by GSSHA and LSTM with the observed data. The evaluation metrics were computed and presented in Table 2.4.

Table 2.4: Prediction performance comparison of GSSHA and LSTM model on selected events.

Event	GSSHA		LSTM	
	RMSE	NSE	RMSE	NSE
Low event	2.508	0.636	2.44	0.655
Moderate event	9.00	0.768	10.02	0.712
High event	16.43	0.506	13.81	0.651
Harvey	182.74	0.710	96.64	0.919

The performances of GSSHA and LSTM were similar at low or moderate rainfall events as shown in Figure 2.9a and 2.9b. However, when it comes to higher precipitation events, LSTM offers prediction that is closer to the observed value and is more robust compared to GSSHA. For Hurricane Harvey, the GSSHA model over-predicted the peak flow by 82.0%, while LSTM under-predicted the peak flow by 15.6%. Moreover, the oscillation of GSSHA prediction depicted in Figure 2.9d indicates possible numerical stability issue or lower than ground truth roughness coefficient were set.

2.6 Discussion

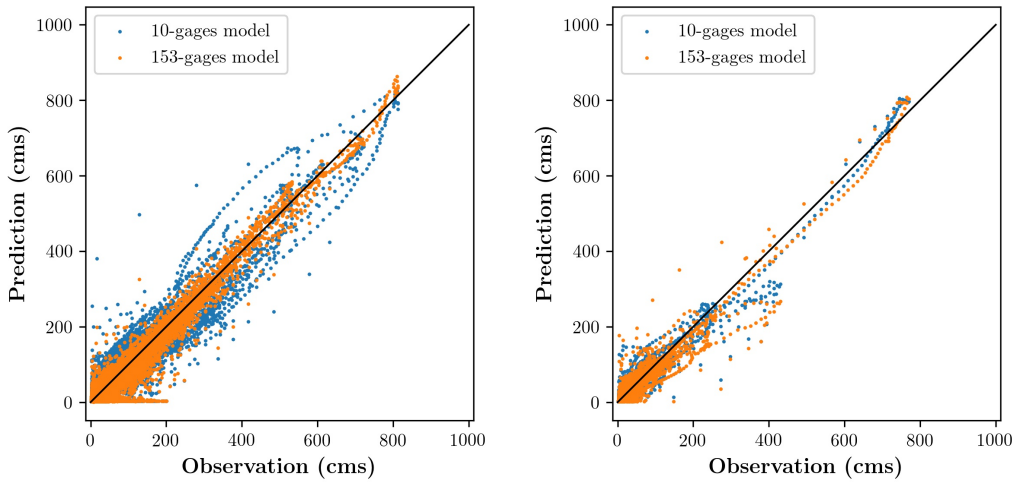
In this chapter, the potential use of Long-Short-Term-Memory networks (LSTM) for RR modelling, using 15-minutes discharge and precipitation data were successfully tested for the first time. In addition, the physical consistency of the LSTM model in terms of its attention distribution on the input space was explored.

The designed numerical tests showed that the LSTM model can accurately predict streamflow given precipitation as the sole input when the scales of test data and training data are identical (interpolation). Furthermore, an LSTM model with the proper regularization and choice of rainfall gauges can even extrapolate well as shown in the prediction of Hurricane Harvey. The performance of the LSTM model was remarkable considering the training data was highly skewed towards base flow rate when little precipitation was presented. As complicated as the LSTM model is, the gauge importance can be evaluated using the weights defined in this study. This is mainly because the gauge readings are measures of the same physics quantity and have similar scale and distribution.

When compared to the process-driven model, GSSHA, the data-driven model is clearly more efficient and robust in terms of inference and calibration. A 20-day event with extreme precipitation can be predicted within a second using LSTM,

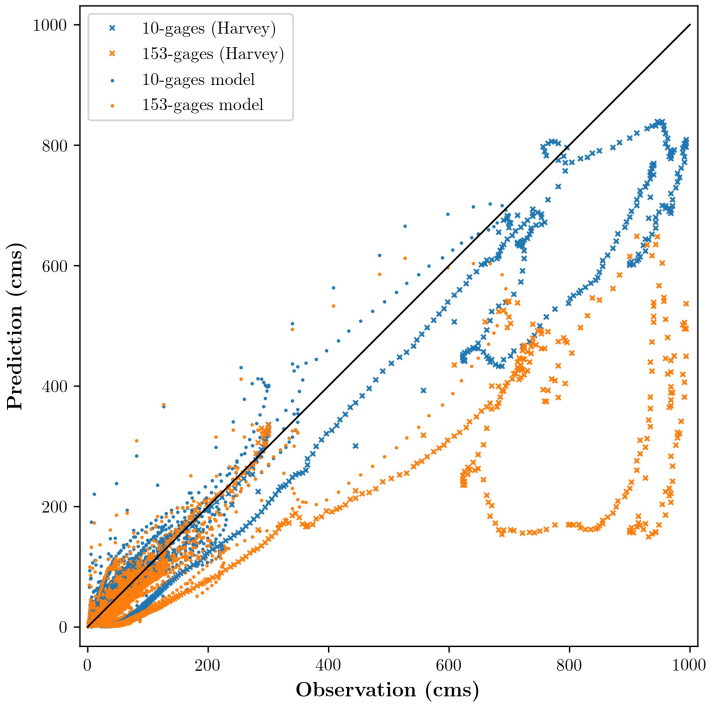
while the same prediction using GSSHA can take hours to a day depending on the topography of the watershed and resolution of the discretization. In this study we found that the calibration of the LSTM network using the 10-years dataset is faster than calibrating GSSHA using a 20-days event. Besides, given that both the data-driven and the process-driven models were not extensively tuned, the LSTM model is more robust and accurate while predicting high precipitation events. Calibration of the process-driven models would be restricted by its prediction efficiency while data-driven models do not have such restrictions.

Finally, it is found that the gauge attention described in this paper can be used as a coarse yet fast metric to select gauges in flood prediction. The experimental result shows that using the gauge attention criterion is better than selecting the closest gauges.



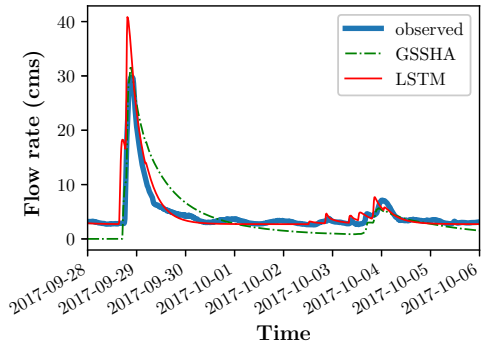
(a) Training set.

(b) Validation set.

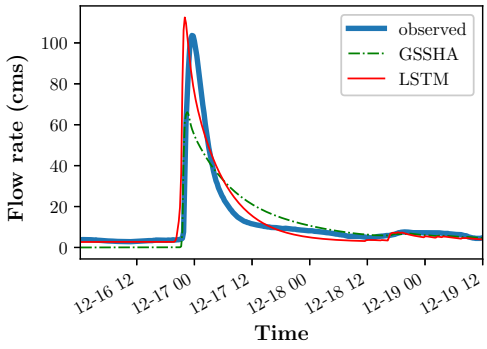


(c) Test set. Note that cross shaped dots are data points from Hurricane Harvey.

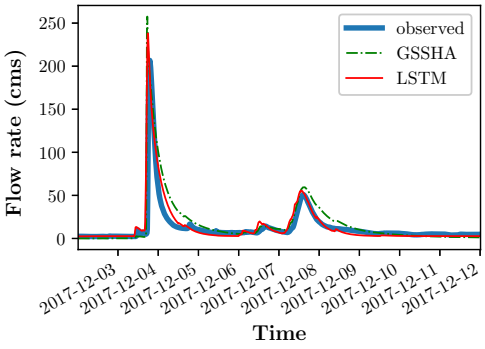
Figure 2.8: Scatter plot of prediction versus observation for 10-gauges and 153-gauges models.



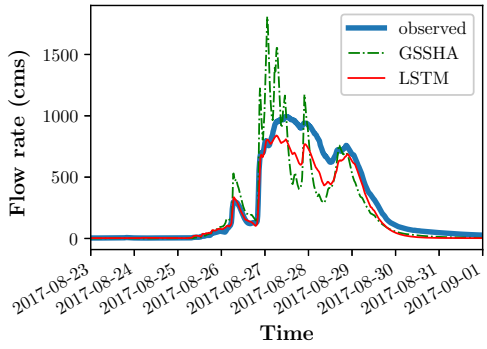
(a) Low rainfall event.



(b) Moderate rainfall event.



(c) High rainfall event.



(d) Hurricane Harvey.

Figure 2.9: Comparison of the ground truth flow rates and predicted flow rates computed by GSSHA and LSTM. The distinct initial gap between prediction of GSSHA model and observation shown in Figure (2.9a) and (2.9b), is due to cold starting GSSHA simulation without an initial condition. Such a gap becomes invisible as event scale increases (Figure (2.9c) and (2.9d)).

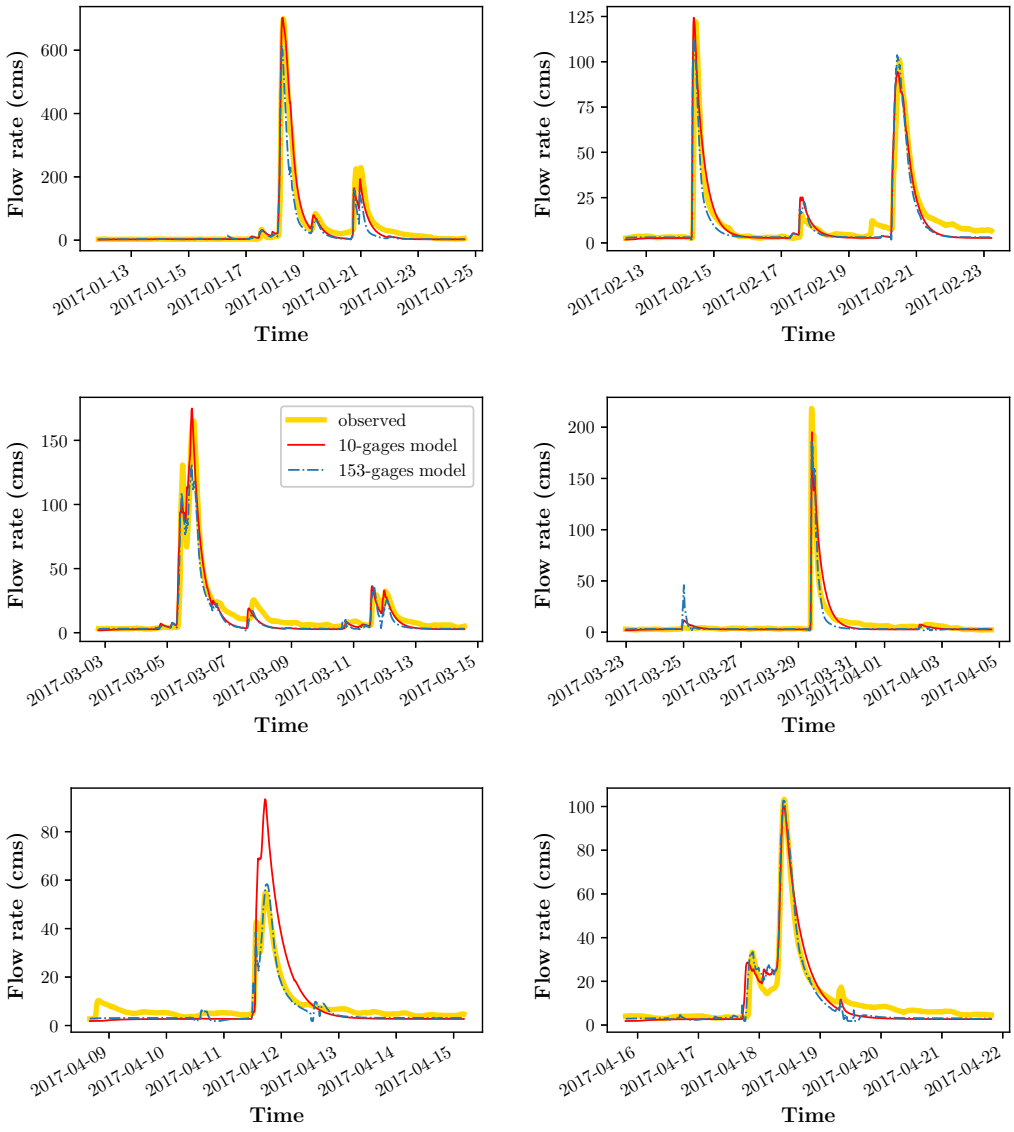


Figure 2.10: Comparison of observation, 10-gauges LSTM, and 153-gauges LSTM

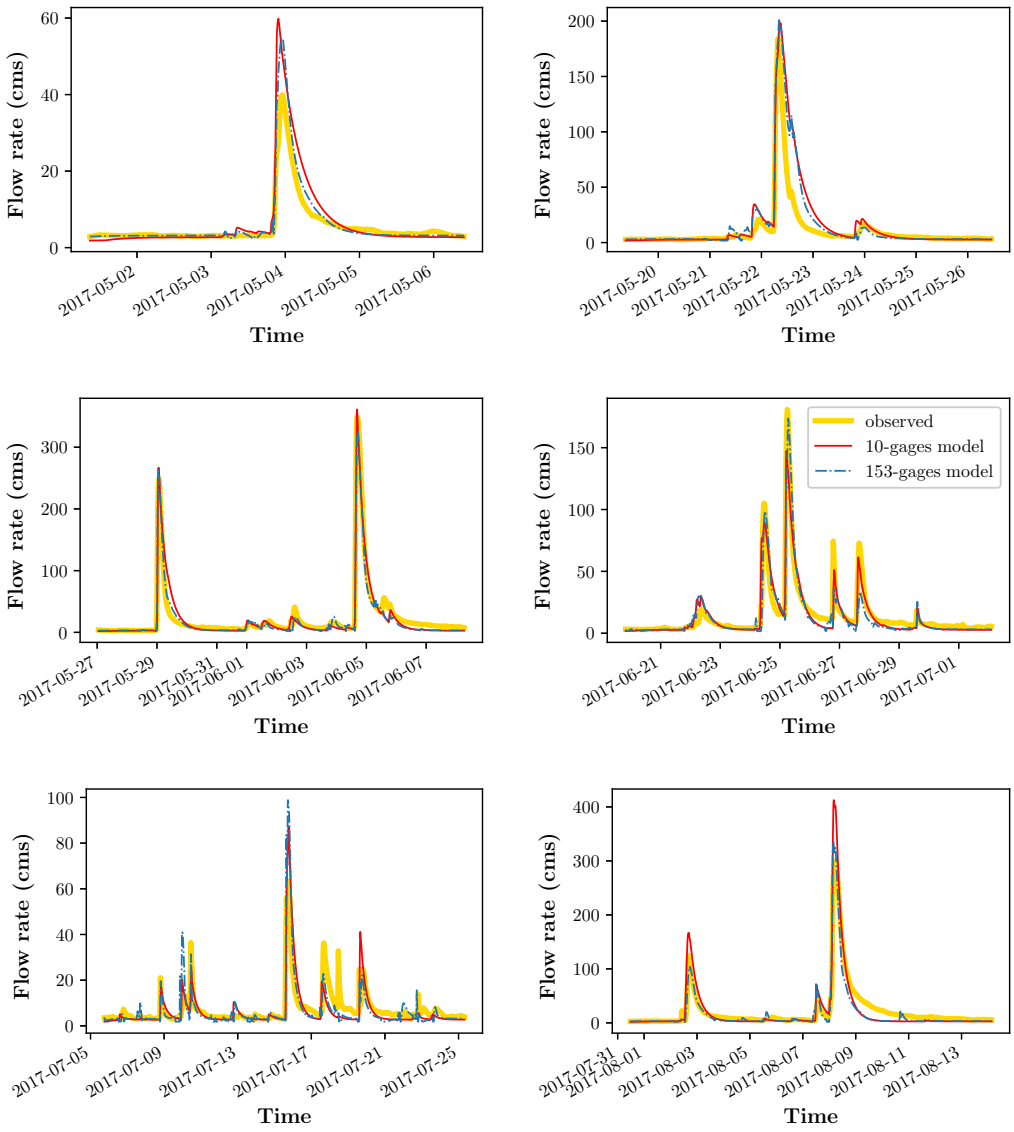


Figure 2.10: (Cont.) Comparison of observation, 10-gauges LSTM, and 153-gauges LSTM

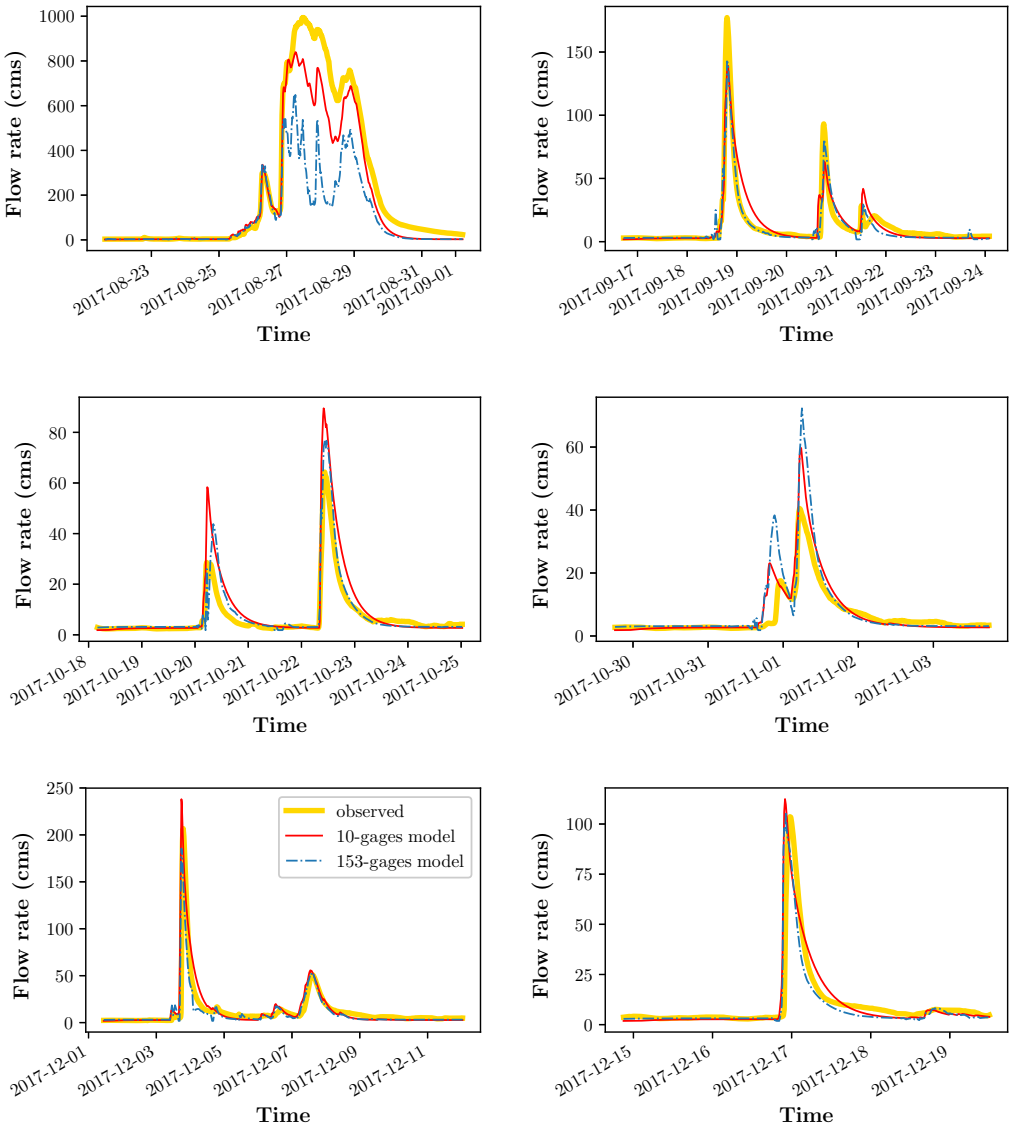


Figure 2.10: (Cont.) Comparison of observation, 10-gauges LSTM, and 153-gauges LSTM. A close look at the lower discharge region would reveal the oscillation of 153-gauges prediction, indicating overfitting.

Chapter 3

RNN architecture for hydrologic modeling

The loops inside the RNN make it capable of capturing long-term dependencies in data. Theoretically, the hidden state inside RNN should be able to preserve the memory of past input data. However, in reality, standard RNN with artificial neurons as hidden units faces vanishing and exploding gradient issues for network training [37]. To this end, long short-term memory (LSTM) networks have been developed by adding cell state and gating mechanisms to the vanilla RNN. The gates within the LSTM network handle the decision process on forgetting or remembering the information by keeping the errors in memory, which avoids error signal decay. In other words, the gates within the LSTM help to preserve states and short-term dependencies over long periods.

The success of the recurrent neural network (RNN) applied to sequential models has motivated groups to pursue RR modeling using RNN. Existing RNN

This chapter is based on published article Li, W., Kiaghadi, A. & Dawson, C. Exploring the best sequence LSTM modeling architecture for flood prediction. Neural Comput & Applic 33, 5571–5580 (2021). https://doi.org/10.1007/s00521-020-05334-3. The author of this dissertation proposed comparing the synced sequence to sequence and the sequence to one RNN rainfall runoff models, implemented the models, and conducted the numerical tests.

based methods generally use either sequence input single output or unsynced sequence input and output architectures. In this chapter, the synced sequence input and output long short-term memory (LSTM) network architecture is compared to existing methods (sequence input single output LSTM) for hydrologic analysis. The proposed architecture is expected to improve RR prediction in terms of accuracy, calibration training time, and computational cost. The key idea is to efficiently learn the long term dependency of runoff on past rainfall history. To be more specific, the architecture utilizes the indigenous ability of the LSTM network to preserve long term memory instead of artificially setting a time window for input data. In this way, the method can effectively avoid losing long term memory of the input, eliminate the calibration of the time window length, and save computational resources. The forward propagation of the network mimics the traditional process-driven methods, and thus resembles the physics interpretation of the RR process. Experiments are conducted using real-world hydrologic data from the Brays Bayou in Houston, Texas. Extensive numerical experimentation clearly validate the effectiveness of the proposed method in terms of various statistical and hydrologic related evaluation metrics. Notably, the experiment shows that some rainfall events could affect the runoff process in the test watershed for at least a week. For fine temporal resolution prediction, this long term effect needs to be carefully handled, and the proposed method is particularly superior in this case.

3.1 RNN Architectures for hydrologic models

A large portion of the aforementioned studies in Table 1.1 could be categorized as time series-forecasting, meaning the flow at previous time steps is within the input vector to the model along with rainfall and/or other variables. Depending on the immediate past observations to make a prediction could be problematic in the case of damage to the observational flow gauge. Severe damages have been reported to

the gauges maintained by the National Oceanic and Atmospheric Administration (NOAA) and the United States Geological Survey (USGS) during hurricanes and severe storms [85]. Some of the damaged gauges have never come back to functionality. Furthermore, future development requires predicting runoff of hypothetical rainfall events with different return periods (scenario-based) as a part of flood management. Such a study would not be possible using the time series-forecasting approach due to the absence of observed discharge data. Thus, in this study, the focus is on regression models for rainfall runoff simulation.

Part of the superiority of RNNs compared to traditional ANNs is due to their sequence regime of operation compared to fixed-size networks. Depending on the application of the network, different architectures, including sequence input and a single output (SISO), single input and sequence output, sequence input and sequence output, and synced sequence input and output (SSIO) can be used (see Figure 3.1 for different structures). While each of these structures are designed for a specific purpose, most of the studies that used the LSTM network for hydrologic modeling have chosen sequence input and single output or sequence input and sequence output architectures(see Table 1.1). These architectures require determining a fixed window size, unlike the SSIO architecture that relies on the LSTM structure itself to capture the long dependencies. Choosing a fixed window size forces LSTM to limit the long-term dependencies into the size of the selected window. On the other hand, the passage of hidden states from previous time steps in the SSIO architecture makes it capable of capturing the long-term dependencies on its own. In other words, there is no need to use LSTM (or more unsophisticated RNN) for hydrologic modeling application if the user desires to use a fixed window regime since the LSTM network does not require fixed size (sequence length) input. Additionally, using the fixed window size approach requires more memory because of the formation of a matrix with a size of batch size $\times$ fixed window size $\times$ number of input variables, compared

to batch size × number of input variables in the SSIO architecture, even though the two input datasets contain the same amount of information. Moreover, extensive knowledge of the watershed response to the rainfall events is required to determine the window size. While the choice of architecture is not very influential when working with a small dataset or coarse time resolution, it could make a significant difference in prediction accuracy, computational time, and storage requirements when the time resolution or size of the study increases. The main goal of this study is to compare these two architectures with regards to prediction performance, computational time, and memory requirements.

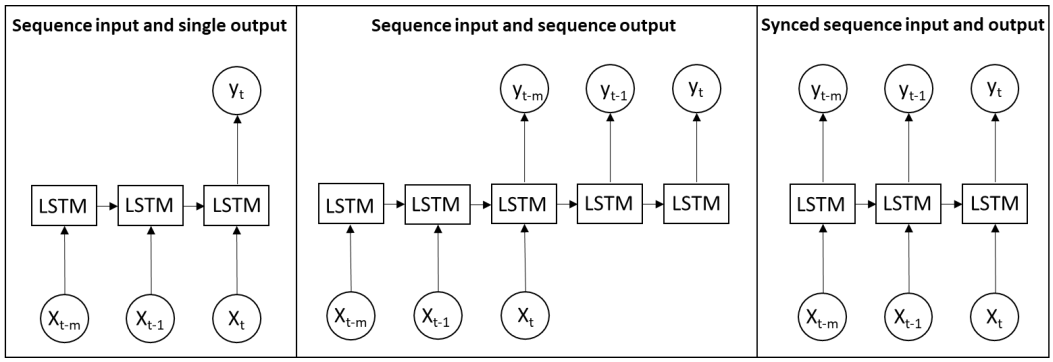


Figure 3.1: Illustration of different LSTM architectures.

3.2 Numerical Experiments

To compare the SISO and SSIO architectures for runoff simulations, numerical experiments were conducted using the dataset from an urban watershed, Brays Bayou, in Houston, Texas (See section 2.1). Since this study is focused on the comparison of different architectures instead of feature selection, data from the rainfall gauges within the Brays Bayou watershed is used as precipitation input. The same train-validation-test data split scheme described in section 2.3 was used. The first step of the comparison study is to train a set of models with the SISO and SSIO architec-

tures respectively.

3.2.1 Synced Sequence Input and Output (SSIO) Model

For the synced sequence input/output model, the training dataset originally consists of 16 columns. Each column is a time series of precipitation or river discharge spanning from 2007 to the end of 2015. Training an RNN on such a long sequence is challenging and problematic. Thus, the training dataset is split into shorter snippets, where each snippet starts from the beginning of a runoff event and ends at a definitive conclusion of a runoff event. The snippets do not necessarily have the same length. During training each snippet is fed into the network as one batch (training proceeds one forward and one backward propagation). Since all the training example starts from a similar meteorological state (river flow at a base flow without precipitation), it is possible to set the initial hidden and cell state to be learnable parameters. At prediction time, the input sequence can either (1) start from a base flow state using the learned initial hidden and cell state, or (2) start at the middle of an event using a hot-start hidden and cell state from the prediction of the first part of the event. From a grid search of hyperparameters, a two-layer LSTM network with 64 hidden LSTM units and zero dropout at each layer has been chosen.

3.2.2 Sequence Input Single Output (SISO) Model

In sequence input single output and un-synced sequence input/output model, the length of the input vector is fixed from the training time. This setup has three potential disadvantages: (1) The excessive need to determine the best window size l_w. In other words, the window size itself becomes an extra hyperparameter and requires tuning. (2) The potential risk of losing long term rainfall history; if l_w is not large enough, the long term effect of rainfall cannot be captured within the time window. For flood prediction at a temporal resolution as high as every 15 min,

there are 96-time steps for a single day. Depending on the size of the watershed, the peak of generated runoff can be observed from a couple of hours to a couple of weeks after the beginning of the event. In this case, if l_w is set to be 100, the network will have trouble predicting a one-week event. (3) Compared to the synced sequence input/output model, the sequence input single output model requires more computation and memory to process the same time series. For instance, if the window size is set to 100, then the input data will be repeated 100 times.

For the sequence input single output model, four different window size (memory length): 48 steps, 96 steps, 192 steps, and 672 steps correspond to 12 hours, one day, two days, and a week were tested respectively. The progressive choices are designed to show the cascade of long term dependency of rainfall-runoff. The tuned hyperparameters of the SISO models are listed in Table 3.1.

Table 3.1: Tuned hyperparameters of SISO LSTM models.

Test name	hidden size	dropout	window size	batch size	early stopping	max epochs
SISO12	64	0.5	48	4096	400	3000
SISO24	64	0.1	96	1024	400	3000
SISO48	128	0.3	192	1024	400	3000
SISO168	128	0	672	1024	200	1000

For all LSTM networks used in this study, a fully connected layer with ReLU activation is added after the last LSTM layer to map the high dimensional hidden state vector h_t^l at each time step t to a scalar output y_t. All LSTM networks are trained with Adam optimizer [72] with the AMSGrad variant [74] and the learning rate is set to 0.00005. The models are implemented using deep learning framework PyTorch [76]. Numerical experiments were conducted on RTX node of Frontera at the Texas Advanced Computing Center (TACC).

It should be noted that the batch size, early stopping epoch, and the maximum epoch number are not tuned to avoid overfitting. The batch size is set to satisfy

the memory requirement of GPU. Early stopping epoch and maximum epoch numbers are set high enough so that the optimization algorithm converges within the limit. The early stopping round and maximum epoch for SISO168 is set lower than other SISO models with shorter window size since training the SISO168 is significantly more time-consuming. It has to be set lower so that training does not exceed the 24-hour time limit for computing jobs on Frontera.

3.2.3 Evaluation Metrics

Extensive evaluation was conducted using both classic and hydrologically relevant metrics. Root Mean Square Error (RMSE), Nash–Sutcliffe Efficiency coefficient (NSE), Mean Absolute Error (MAE) are used to measure the model performance. While RMSE could provide valuable information on the model performance, breaking it down to bias, amplitude error, and the phase error could provide more specific details on the source of error. The following equations express the RMSE decomposition [86]:

$$RMSE = \sqrt{\frac{\sum_{k=1}^{N}(Q_{M,k} - Q_{O,k})^2}{N}} \tag{3.1}$$

$$RMSE^2 = bias^2 + SD_{bias}^2 + DISP^2 \tag{3.2}$$

$$bias = \frac{1}{N}\sum_{k=1}^{N}(Q_{M,k} - Q_{O,k}) \tag{3.3}$$

$$SD_{bias} = \sigma_M - \sigma_O \tag{3.4}$$

$$DISP = \sqrt{2\sigma_M\sigma_O(1 - \rho_{O,M})} \tag{3.5}$$

where $Q_{M,k}$ and $Q_{O,k}$ are the modeled and observed discharges at time step t, respectively; N is the total number of time steps; SD_{bias} is the amplitude error; $DISP$ is the absolute value of phase error where a non zero value indicates the phase of modeled discharge lags or leads the observed one; σ_M and σ_O are the standard deviation of modeled and observed discharges, respectively; and $\rho_{O,M}$ is the correlation coefficient between the observed and modeled discharges.

To exploit the hydrologic context, hydrologic relevant metrics that evaluate overall water balance, vertical redistribution, and temporal redistribution could be used as diagnostic tools. Some of these metrics are derived from the concept of the flow duration curve. FDC is defined as the relationship between a given discharge value and the percentage of time that this value was exceeded. The concept of probability distributions cannot be applied in RR process due to the existing correlation among discharges in successive time and the effect of seasonality. FDC could be considered as the complement of the cumulative distribution function [87]. In this dissertation, the following hydrologic relevant metrics were used:

$$FMS = \frac{(\log Q_{M,0.2} - \log Q_{M,0.7}) - (\log Q_{O,0.2} - \log Q_{O,0.7})}{(\log Q_{O,0.2} - \log Q_{O,0.7})} \tag{3.6}$$

$$FHV = \frac{\sum_{i=1}^{H}(Q_{M,i} - Q_{O,i})}{\sum_{i=1}^{H} Q_{O,i}} \tag{3.7}$$

$$FLV = \frac{\sum_{j=1}^{L}(\log Q_{M,j} - \log Q_{M,min}) - (\log Q_{O,j} - \log Q_{O,min})}{\sum_{j=1}^{L}(\log Q_{O,j} - \log Q_{O,min})} \tag{3.8}$$

$$FMM = \frac{\log Q_{M,median} - \log Q_{O,median}}{\log Q_{O,median}} \tag{3.9}$$

$$EQ_{Peak} = \frac{Q_{M,peak} - Q_{O,peak}}{Q_{O,peak}} \tag{3.10}$$

$$ET_{Peak} = |T_{M,peak} - T_{O,peak}| \tag{3.11}$$

where FMS: bias in flow FDC midsegment slope which evaluates the vertical redistribution, $Q_{0.2}$ and $Q_{0.7}$ are discharges associated with the exceedance probabilities of 20% and 70% (80 percentile and 30 percentile respectively), FHV is the bias in FDC high-segment volume (top 2%), H is the number of runoff indices corresponding to discharges with exceedance probabilities smaller than 2%, FLV is the bias in FDC low-segment volume that evaluates the long-term baseflow, L is the number of runoff indices corresponding to discharges with exceedance probabilities between 70% and 100%, FMM is the bias in the median runoff, EQ_{Peak} is the error of peak runoff, and ET_{Peak} is the error of time to peak runoff.

3.3 Results

The prediction performance is shown in Table 3.2, 3.3, and 3.4. Since the test dataset has an unprecedented event, Hurricane Harvey, where flooding was so severe that inter-basin flow was observed, both the evaluation metrics of the entire test set and the ones excluding Harvey are reported.

Among all the tested methods, the synced sequence input and output model (SSIO) has the best overall performance. It leads all the methods in terms of NSE, MAE, RMSE, and $\rho_{O,M}$. As the length of time window increases, better overall prediction accuracy is observed for the sequence input and single output models (SISO). NSE, MAE, RMSE, and $\rho_{O,M}$ show a clear progressive improvement as the time window increases from 12 to 168 hours. This phenomenon suggests that the runoff at the studied watershed indeed has a long term dependency on past rainfall

Table 3.2: Statistical metrics of predictions.

Test name	Dataset	NSE	MAE	RMSE
SSIO	Test	0.943	132.7	614.9
	Test no Harvey	0.936	72.0	242.0
SISO 12	Test	0.803	250.0	1145.0
	Test no Harvey	0.887	146.9	322.7
SISO 24	Test	0.898	191.5	822.4
	Test no Harvey	0.921	114.9	269.5
SISO 48	Test	0.928	163.0	694.1
	Test no Harvey	0.914	96.2	281.5
SISO 168	Test	0.935	154.9	655.5
	Test no Harvey	0.934	96.0	245.8

Table 3.3: RMSE decomposition of predictions.

Test name	Dataset	RMSE	SDbias	$\rho_{O,M}$	DISP	Bias
SSIO	Test	614.9	421.0	0.982	443.4	-65.32
	Test no Harvey	242.0	-47.8	0.971	237.1	-5.42
SISO 12	Test	1145.0	768.1	0.923	847.4	-54.96
	Test no Harvey	322.7	-20.4	0.946	318.5	47.28
SISO 24	Test	822.4	541.6	0.964	617.3	-43.93
	Test no Harvey	269.5	-25.2	0.962	266.4	32.49
SISO 48	Test	694.1	445.1	0.974	531.3	-38.48
	Test no Harvey	281.5	-80.4	0.964	268.9	22.71
SISO 168	Test	655.5	362.1	0.974	545.8	-25.36
	Test no Harvey	245.8	-11.6	0.968	243.7	30.08

history; thus, architectures that could preserve long term memory are required to model flood on a fine time scale. Phase error (DISP) was the most significant contributor to the RMSE, followed by amplitude error. Table 3.4 shows SSIO has the best prediction performance in terms of hydrologic related evaluation metrics. SSIO has the best bias in median runoff FMM among all the methods. The better performance of the SSIO method in predicting the median discharge could also be seen in Figure 3.2A.

Figure 3.2B depicts the FDC for all test cases. The vertical and horizontal axes show the exceedance rates and their corresponding discharge rates (cfs or

Table 3.4: Hydrologic relevant metrics of predictions.

Test Name	Dataset	FMS	FHV	FLV	FMM
SSIO	Test	-21.1%	-16.5%	-64.7%	2.6%
	Test no Harvey	-20.2%	-0.8%	-67.0%	2.4%
SISO 12	Test	-88.0%	-30.3%	-250.9%	-9.8%
	Test no Harvey	-87.3%	-5.0%	-256.0%	-10.1%
SISO 24	Test	-81.0%	-21.6%	-198.7%	-6.7%
	Test no Harvey	-80.9%	-3.1%	-203.0%	-7.0%
SISO 48	Test	-48.9%	-15.9%	-128.4%	-2.6%
	Test no Harvey	-50.1%	1.4%	-131.6%	-2.8%
SISO 168	Test	-52.0%	-16.3%	-186.0%	-4.9%
	Test no Harvey	-52.5%	-6.2%	-190.2%	-5.2%

ft^3/s), respectively. An exceedance rate for a specific discharge rate means how much percentages of all flows have a value greater than that rate. Within the FDC (Figure 3.2B) low-segment region (70–100%), the SSIO model also shows superior performance. It has significantly lower FLV compared to the other methods. Note that for this metric excluding Hurricane Harvey does not change the result significantly, because most of the data points in Hurricane Harvey are outside this region due to large discharge rates during this event. We believe that the SSIO model benefits from setting the initial hidden state to be learnable during training. The setting essentially forces the network to learn the base flow condition from data.

Figure 3.2A also provides a better illustration of the FDC mid-segment region (corresponding bias: FMS) by showing the 25, 50, and 75 percentiles of discharge values for all test datasets. Here again, SSIO showed the most similar pattern to the observed data, followed by SISO168 and 48. Within FDC high-segment region, SSIO has similar FHV with SISO48 and SISO168 on the entire dataset. If Harvey is excluded, the SSIO has the smallest absolute value of FHV. This suggests its superior prediction performance within the high flow region, which is possibly due to its ability to preserve the entire memory of runoff events with a long duration. In flood risk management, we are particularly interested in predicting the magnitude and the

Test	Dataset	NSE	MAE	RMSE	ETPeak	EQPeak (%)
SSIO	2016 Eventa	0.988	436.1	668.5	0.5	1.8
	Harvey	0.922	1891.6	3105.7	5.25	-2.5
SISO 12	2016 Event	0.885	1141.1	2111.2	0.75	5.6
	Harvey	0.708	3239.1	6026.3	4	-16.5
SISO 24	2016 Event	0.973	724.9	1020.7	0.5	3.3
	Harvey	0.854	2415.0	4264.3	5.75	-11.3
SISO 48	2016 Event	0.987	464.7	707.2	2	3.4
	Harvey	0.902	2101.5	3486.7	5.25	-13.3
SISO 168	2016 Event	0.988	434.5	694.9	4.25	-2.3
	Harvey	0.910	1863.9	3337.6	5.5	21.1

Table 3.5: Prediction performance for historic rainfall events in Brays Bayou.

time of the peak flow for runoff events. Evaluation metrics for two demonstrative events are shown in Table 3.5. In addition to Hurricane Harvey, the prediction of the Tax Day Flood event (2016), which is part of the validation dataset, is also investigated as the second-largest event in the history of Brays Bayou. This particular storm was chosen because the peak flow on this event was very similar to Harvey. However, the duration and RR behavior of Harvey were drastically different from Tax Day Flood or any other historical events.

Even though the 2016 event is in the validation dataset and used for hyperparameter tuning, we can still see the superior performance of the SSIO method on FDC high-segment regions. More importantly, from the prediction evaluation metrics (see Table 3.4 and Figure 3.2), it can be concluded that the developed LSTM models could predict a very big event (with flow rates as high as 27,200 cfs) with acceptable error. Thus, the evaluation metrics show that the data-driven method can precisely predict runoff of a historic rainfall event. From the NSE/MAE/RMSE metrics in Table 3.5, it can be concluded that the SSIO and SISO168 model have similar overall prediction performance on both the Tax Day Flood and Harvey. However, on Harvey, SSIO underpredicted the peak flow by 2.5%, while SISO168 overpredicted it by 21.1%. It should be noted that Hurricane Harvey was an unseen

event in history (flow rates as high as 35,000 cfs), so the models have to extrapolate to predict its discharge rates. Figure 3.3 shows the LSTM model predictions for Harvey compared to the observed values. The ET_{Peak} metric and Figure 3.3 suggested that all the methods were able to identify the correct pattern of the disastrous runoff event (except SISO 12). But once again, progressively improved performance was observed as the lengths of input memory increases. As the time window size of SISO reaches a week, prediction performance becomes similar to the SSIO model.

To evaluate the LSTM models' computational time and costs, the training times and stopping epochs were recorded. It is found that for SISO architectures with short time windows (12 hours or one day), training of one epoch takes a shorter time (3 seconds and 8 seconds, respectively) compared to the SSIO model (21 seconds), because load balancing is easier for SISO models (training sequences are of uniform length) than SSIO models (training sequences are of variant length in our study). As the window size increases to two days and a week, the training time per epoch (25 seconds and 61 seconds for SISO48 and SISO168, respectively) will be longer than the SSIO model. Note that this is not a strict performance timing test, and training time per epoch depends on the number of parameters of the network, batch size, and training strategy. However, the trend of SISO training time suggests its inefficiency. The training process of SISO architecture is more scalable because it takes input sequences with uniform length and load balancing of parallel computation is easier. However, as the length of time window increases, the additional computation is going to offset the benefit of scalability eventually. Particularly, during the training of SISO168, the early stopping and maximum training epochs had to be reduced so that the training could be completed within the 24-hour limit on TACC. Moreover, the longer time window leads to a larger memory requirement, which restricts the training batch size. For example, setting the batch size to 4096 for SISO168 will exceed the GPU memory limit.

3.4 Discussion

This chapter shows the superiority of synced sequence input and output (SSIO) LSTM architecture for hydrologic analysis over existing methods that use sequence input single output (SISO) architecture. The experimental results from real-world hydrologic data validated that SSIO architecture is not only more accurate but also consumes less computational resources than SISO. The advantage of SSIO architecture is especially significant under scenarios where fine temporal resolution is required. Hydrologic related evaluation metrics show that the SSIO method has better performance from the hydrologic perspective. Moreover, the numerical results suggest that particular rainfall events could affect the runoff process in the test watershed for at least a week. Thus, careful treatment of the long term dependency is essential for fine temporal resolution hydrologic modeling. To this end, this study suggests that SSIO is a more suitable architecture for RNN-based hydrologic modeling.

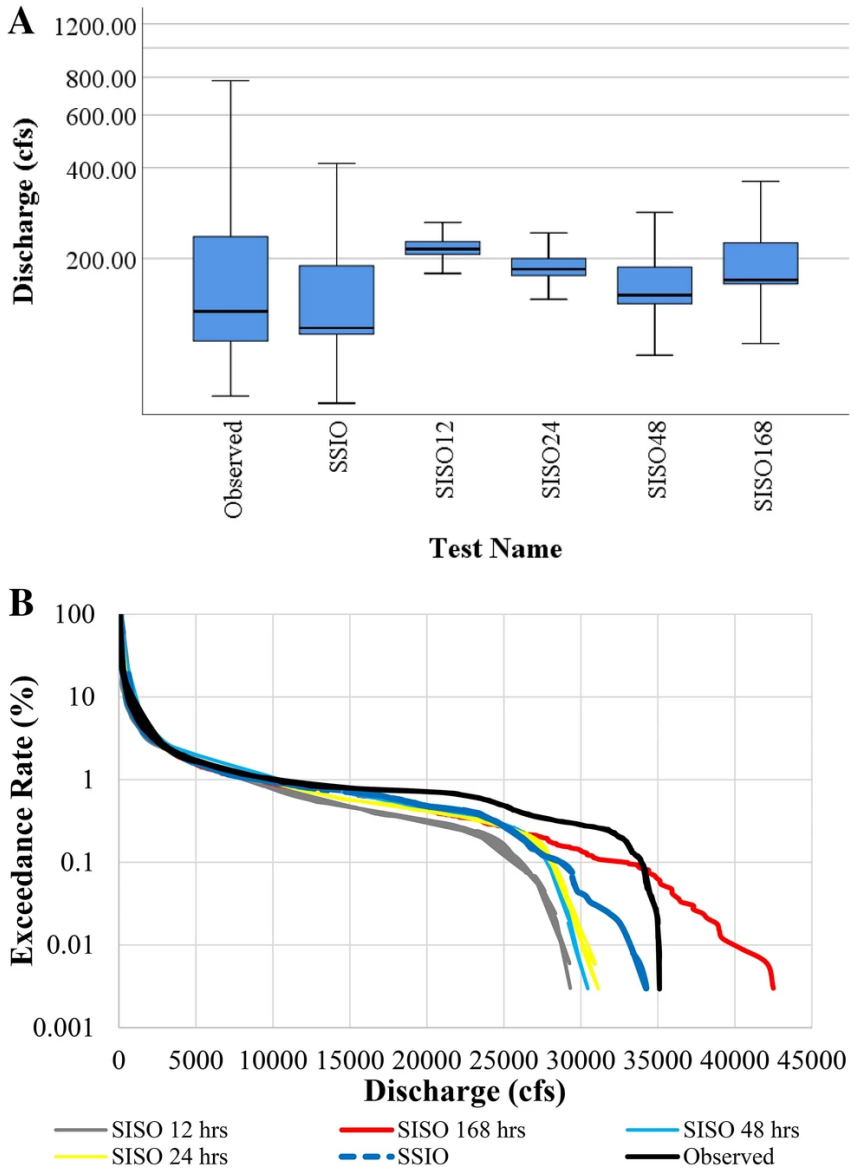


Figure 3.2: A) Box plot (outliers are not shown) and B) flow duration curve (FDC) of test cases for all tested architectures and the observed discharge.

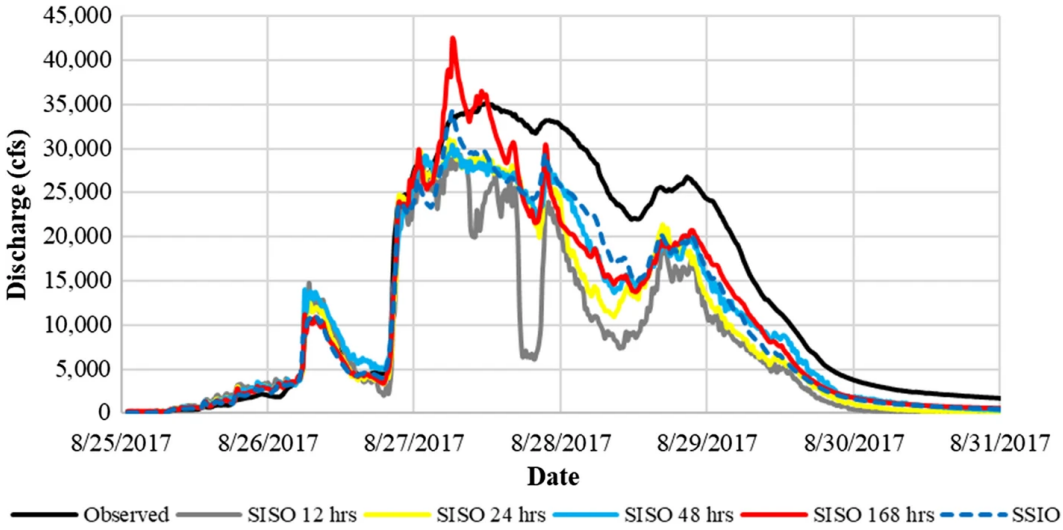


Figure 3.3: LSTM model predictions for Hurricane Harvey compared to the observed values.

Chapter 4

Coupled model for compound flooding

Accurate simulation of compound flood events in coastal regions requires considering two mechanisms, rainfall runoff and storm surge. Neglecting any of them can cause significant errors in local water surface elevation estimates. Although much effort has been spent towards coupling the two mechanisms, there still lacks an efficient two-way coupled compound flooding model. In this chapter, a data-driven hydrologic model using a deep recurrent neural network that takes downstream water level into account is presented. Then the RNN model is dynamically coupled with an ocean circulation model, specifically, the Advanced Circulation (ADCIRC) model, for compound flooding simulation. The improvement from adding downstream water level information to the RNN is shown by an ablation study. Later, by coupling the neural network and ADCIRC, a deep learning/computational hybrid compound flood simulation model was developed and tested for the Galveston Bay region in Texas.

4.1 Study Area and Data Acquisition

The Houston Ship Channel-Galveston Bay (HSC-GB) estuarine system, located in the south eastern part of Texas, was chosen as the testbed for this study. The potential of flooding due to various types of flooding sources including inland runoff, wind driven storm surge, and sea level rise makes this region a suitable testbed. Some of the recent severe flooding events within the study area showed some elements of compound flooding in which two or more sources of flooding happened at the same time or within a short period of time from each other. For instance, Hurricane Ike (2008) and Harvey (2017) were identified as compound events [10]. As noted before, the focus of this study is not on simulating a flooding event that affects the entire estuarine system but rather on developing a coupling framework. Thus, and to reduce the computational cost of this study, only one watershed was selected to demonstrate the coupling framework. Among the watersheds draining into the HSC-GB system, Brays Bayou watershed (see Figure 4.1) is the most developed one with the longest history of severe flooding and inundation. Additionally, previous works on the use of machine learning algorithms in predicting runoff just by using rainfall as the input file showed promising results in Brays Bayou watershed [1, 2].

Within Brays Bayou watershed, there are 15 rainfall gauges maintained by the Harris County Flood Control District (HCFCD). Hourly precipitation intensity data from 2007-2017 was used for the training/validation/test periods. Freshwater flow data was downloaded from the United States Geological Survey (USGS) gage 08075000 (fresh gage). Changes in water surface elevation over time were calculated using the gage height data from the USGS gage 08075110 at MLK Jr Blvd (tidal gage). Water surface elevation at Morgan's point was obtained from a gage maintained by the National Oceanic and Atmospheric Administration (NOAA). To validate the storm surge predictions of the coupled model, four validation points are selected (see Figure 4.1).

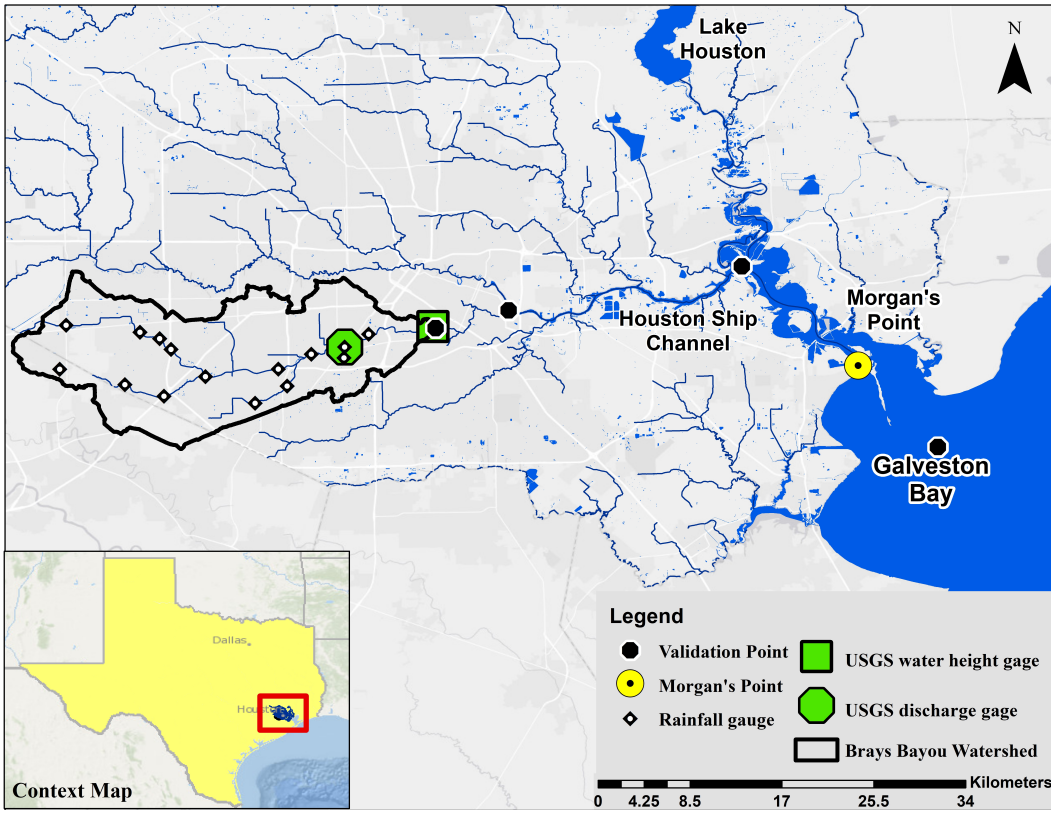


Figure 4.1: Study Area: The Houston Ship Channel-Galveston Bay (HSC-GB) estuarine system. The USGS water height gage is USGS 08075000 and USGS discharge gage is USGS 08075110 at MLK Jr Blvd.

4.2 One-way coupled hydrologic model with deep recurrent neural network

When the coupled physics-driven hydrologic model, which provides the flux boundary conditions, is being replaced with a data-driven model, the feedback from the physical coastal model could be missed. This is mainly due to the lack of downstream boundary conditions in the machine learning model which could be resolved by using the output from the primary model as an input for the data-driven model. Thus, to feed the data-driven model with the storm surge information, the downstream water surface elevation needs to be added to the input of the neural network.

However, choosing the right location within the primary model domain to provide such feedback is critical. When coupling physics-driven hydrologic models with the coastal ones this feedback is generally forced as boundary conditions at/near the outlet of the watershed. A similar choice for a data-driven model may pose a serious problem due to a statistical reason: the strong correlation between water level and river discharge at the same location. Flow measurement at a specific location (i.e., USGS flow gage) has been done by converting the water level to a discharge using a stage-discharge curve. If the river stage is being used to predict the river streamflow at the exact same gage, a data-driven model tends to learn the rating curve instead of correlation between other variables. As a result, the chosen point/location should be far enough from the coupling point and other flow boundaries to only represent the storm surge component of the flood. On the other hand, the chosen point should not be too far away from the coupling location that it does not reflect the timing and pattern of the storm surge at the flow boundary. The difference in bathymetry and bed cover could cause changes in the magnitude and timing of a storm surge from the chosen point to the flow boundary. Within our study area, Galveston Bay is very different from the HSC riverine and shallow bay regions. Thus, considering all of the aforementioned deliberations, Morgan's Point

(see Figure 4.1 for the location), which is the mouth of the HSC to Galveston Bay was selected as the representing location for the storm surge and tides that might affect flows and water surface elevations in Brays Bayou.

4.2.1 Numerical experiment

To show the compound effect of coastal flooding in the area, an ablation study was conducted where three different input combinations to the RNN models were developed: (1) Precipitation model: input consists exclusively of precipitation from rain gages; (2) Water level model - Morgans Point: input consists exclusively of downstream water level at Morgans Point; (3) Coupled RNN model - Morgans Point: input consists of both precipitation and downstream water level data at Morgans Point. By comparing scenario 1 and 3, we can understand the informational improvement of adding the feature of downstream water level to the deep learning model. Moreover, scenario 2 could provide an indicator of the correlation between the added feature (downstream water level) and the target (river discharge/stage). Furthermore, two groups of RNN runoff models were trained to predict two different targets respectively: the river streamflow at fresh gage (USGS 08075000) and river stage at a tidal USGS gage (USGS 08075110, see Figure 4.1). Although the physical distance between the two gauges is within 7 kilometers, the impact of compound flooding on the streamflow and stream stage at the two locations can be drastically different.

The deep recurrent neural network used to model rainfall runoff in this study is a synced sequence input and output Long short-term memory (LSTM) network proposed in [1] (described in Chapter 2). As shown in Figure 2.3, the network has two LSTM layers and a fully connected layer with rectified linear unit (ReLU) as activation. Deep learning framework PyTorch [76] was used to implement the proposed neural network. Compared to other RNN architectures used in hydrologic

modelling (e.g. sequence input and single output), this architecture not only predicts rainfall runoff with better accuracy using less computational resources, but also enables time-stepping at prediction time without extra data preprocessing [2].

Each model was calibrated and tuned using the same training/validation/test split scheme as shown in Table 4.1. Data from 2007 to 2015 is used for training the networks. Data from the entire 2016 is used for validation and hyperparameter tuning. Data from the entire 2017 is used as the hold-out test dataset to evaluate the performance of the model. During training, the loss function of the models is the mean squared error (MSE). To minimize the loss function, Adam optimizer [72] was used. Hyperparameters including learning rate, number of LSTM layers, number of LSTM units per layer, and dropout rate are tuned using a grid search approach. The combinations with best validation Nash–Sutcliffe Efficiency coefficient (NSE) are selected and shown in Table 4.1. Both the input and output variables of the network are scaled to the range $[0, 1] \in \mathbb{R}$.

Table 4.1: Training, validation, and test data split scheme.

Dataset	Period (streamflow at fresh gage)	Period (stage at tidal gage)
Training	10/01/2007 - 11/27/2015	10/01/2007 - 01/15/2016
Validation	11/27/2015 - 12/10/2016	01/15/2016 - 01/04/2017
Hold-out Test	12/10/2016 - 12/31/2017	01/04/2017 - 12/31/2017

A common challenge in training machine learning models based on real world data is the missing data issue. In this study, RNN models were trained with gaged observations. Some of the missing data (e.g. random missing discharge gage readings) can be padded by interpolation or simply eliminated. Some, however, cannot be easily fixed by any interpolation technique. A large trunk of water level readings at Morgans' point during Hurricane Ike (2008) is missing. Excluding Hurricane Ike, as one of the only available compound flooding events during the training period, could lead to losing valuable training dataset. To fill this data gap, a hindcast of

Hurricane Ike was run using a calibrated validated ADCIRC model for the study area. Since the RNN model was developed for rainfall runoff prediction under precipitation, it is reasonable to exclude base flow data for evaluation (though they were included in training dataset). The majority of records are base flow without any precipitation. Thus, data points with river streamflow (or stream stage) lower than 70-percentile (30% exceedance rate) are excluded from the evaluation. Note that both evaluation and training use the observed input data. Practically, such information is not directly available at prediction time, and precipitation and downstream water level forecasts are needed; e.g. water level at Morgans Point is supplied by ADCIRC. To evaluate the trained RNN models, statistical metrics root mean square error (RMSE) and NSE are used. In addition, flow duration curve (FDC) and several hydrologic metrics including FMS, FHV, FLV, FMM, and EQ_{Peak} (see Section 3.2.3) are used to further analyze the performance of the trained RNN models.

4.2.2 Results

The prediction performance of river streamflow at fresh gage and river stage at tidal gage are shown in Table 4.2, 4.3 and Table 4.4, 4.5 respectively. For both targets, it is found that the coupled RNN model outperforms the precipitation model due to the additional information of downstream water level, even though the water level itself does not provide sufficient information to infer the streamflow or the river stage. The NSE of the best water level models on the validation dataset are 0.5348 for the discharge at fresh gage and 0.1601 for the river stage at tidal gage. Note that although evaluation metrics of training, validation, and test datasets are computed and presented in Tables 4.2 - 4.5, one should only rely on the test dataset for performance evaluation since it is the holdout dataset that is not visible to the model throughout the training/validation process.

Moreover, the evaluation metrics indicate that the downstream water level

Table 4.2: Statistical metrics of RNN prediction of river streamflow at fresh gage.

Dataset	**Model**	**NSE**	**RMSE (m3/s)**
Test	Precipitation RNN	0.9623	21.70
	Coupled RNN	0.9882	12.13
Validation	Precipitation RNN	0.9729	8.50
	Coupled RNN	0.9729	8.50
Training	Precipitation RNN	0.9600	10.24
	Coupled RNN	0.9595	10.30

Table 4.3: Hydrologic relevant metrics of RNN prediction of river streamflow at fresh gage.

Dataset	Model	FMS	FHV	FLV	FMM	EQ Peak
Test	Precipitation RNN	51.2%	-12.85%	-85.8%	-5.53%	-13.42%
	Coupled RNN	46.1%	0.25%	-74.1%	-7.26%	-4.92%
Validation	Precipitation RNN	70.6%	-2.72%	-93.2%	-4.65%	2.90%
	Coupled RNN	69.8%	-1.20%	-91.6%	-6.55%	1.58%
Training	Precipitation RNN	33.2%	1.68%	-29.5%	-2.41%	0.187%
	Coupled RNN	33.0%	3.28%	-18.6%	-3.92%	1.23%

affects streamflow/stage at fresh/tidal gages differently. When discharge at the fresh gage is the target of the machine learning model, adding downstream water level to the input does not significantly change evaluation results on the training or validation dataset. However, on the hold-out test dataset, the coupled RNN model has superior prediction performance over the precipitation RNN model. By adding the additional information, NSE is improved from 0.9623 to 0.9882. The coupled RNN model also has significantly lower test FHV but similar FMS and FMM compared to the precipitation RNN.

Figure 4.2 shows the flow duration curves of the predictions and observations during the test period at the fresh gage. The FDC (Figure 4.2) and hydrologic metrics such as FMS suggest that both models match the gage measurements well in the FDC high segment (high streamflow); and both are relatively inaccurate in the medium to low segment. However, within the test dataset, the coupled RNN

Table 4.4: Statistical metrics of RNN prediction of river stage at tidal gage.

Dataset	Model	NSE	RMSE (m)
Test	Precipitation RNN	0.8323	0.513
	Coupled RNN	0.9261	0.341
Validation	Precipitation RNN	0.7322	0.334
	Coupled RNN	0.9548	0.137
Training	Precipitation RNN	0.7569	0.361
	Coupled RNN	0.9222	0.204

Table 4.5: Hydrologic relevant metrics of RNN prediction of river stage at tidal gage.

Dataset	Model	FMS	FHV	FLV	FMM	EQ Peak
Test	Precipitation RNN	43.01%	-14.94%	185.2%	-103.26%	-3.13%
	Coupled RNN	-2.34%	-15.74%	460.1%	-2.35%	-4.03%
Validation	Precipitation RNN	96.28%	3.81%	258.8%	-73.96%	4.77%
	Coupled RNN	-1.16%	-1.75%	346.7%	0.59%	1.13%
Training	Precipitation RNN	80.71%	-2.50%	854.0%	-103.12%	1.13%
	Coupled RNN	1.56%	-2.07%	1781.6%	-0.23%	-1.71%

model is particularly accurate in the FDC high segment. It has significantly better FHV and peak runoff error.

The peak of the test dataset occurred during Hurricane Harvey, the runoff streamflow is shown in Figure 4.3a. From the figure, it can be seen that the precipitation model not only under-predicted the peak by 13.42%, but also missed the third runoff peak on the evening of August 28 by a large margin. The coupled RNN prediction follows the recorded streamflow closely. The only visible discrepancy is the -4.92% error of the peak runoff. The result shows that the RNN model can generalize better by the adding downstream water level.

On the other hand, Table 4.4 and 4.5 shows that the predicted river stage at tidal gage responds to the additional information differently. The coupled model has better statistical evaluation metrics in training, validation, and test dataset. The consistent improvements indicate that runoff at this location is more correlated with

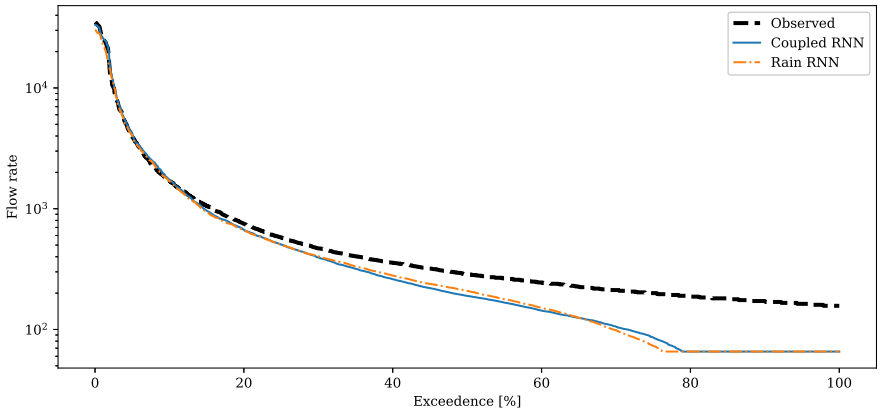
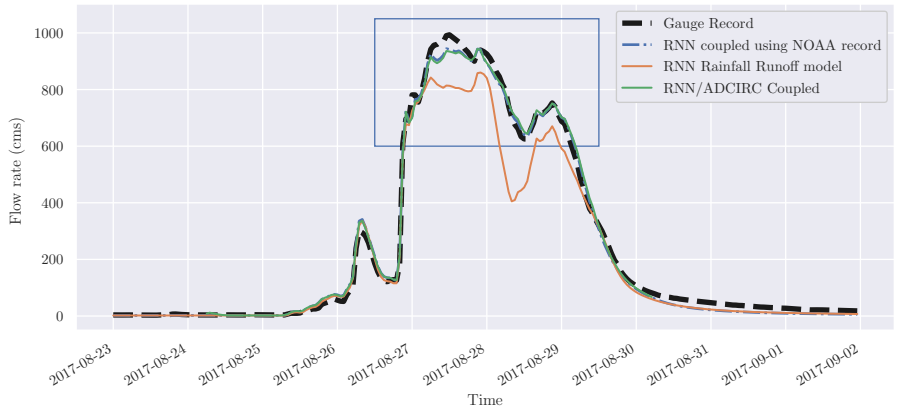


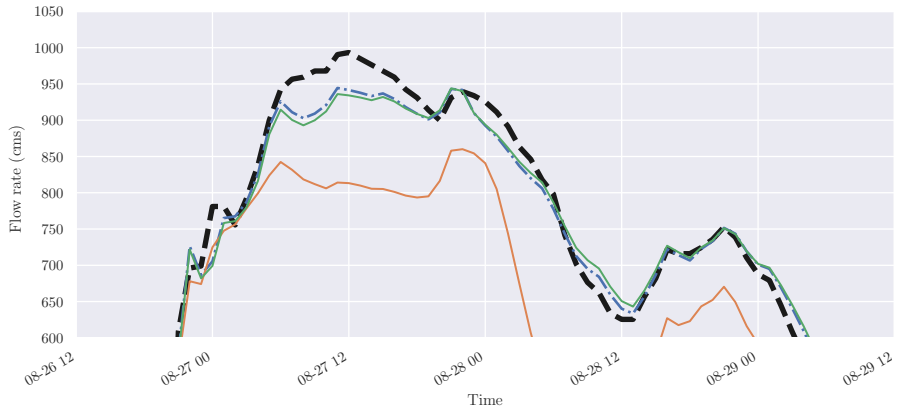
Figure 4.2: FDC of streamflow at fresh Gage.

the downstream water level information. Using the concept of flow duration curve, Figure 4.4 shows the stage-exceedance relationship. The corresponding hydrologic evaluation metrics were also analyzed (see Table 4.5). Lower FMS and FMM were observed in training, validation, and test dataset. The two models have similar FHV and peak stage error in all three datasets. As shown in Figure 4.4, the precipitation RNN significantly deviates from the observation in the medium-low stage segment. However, with the help of the water level at Morgans Point, the coupled RNN model performs better in this segment.

Taking a closer look at the river stage at tidal gage during Hurricane Harvey (Figure 4.5), we will find that both models could accurately capture the first and second peaks. But both of them overestimated the runoff speed after the stage peak on August 27. Without access to the downstream information, the precipitation model made larger errors in prediction after August 29 than the coupled model.



(a)



(b)

Figure 4.3: Prediction of river discharge at fresh gage during Hurricane Harvey. Note that the RNN coupled prediction is very similar to the RNN/ADCIRC dynamic coupled prediction. For better visualization, the section within the blue rectangular in (a) is zoomed in and shown in (b).

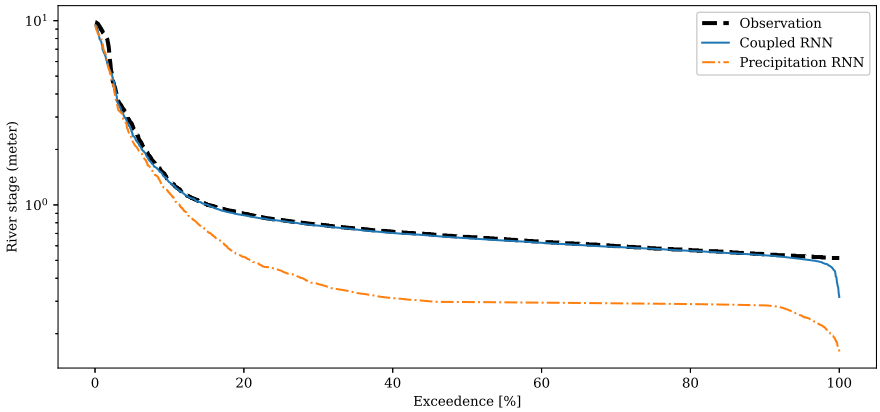


Figure 4.4: Stage duration curve of river stage at tidal gage.

4.3 Dynamic coupling of data-driven hydrologic model and physics-driven hydrodynamic model

4.3.1 ADCIRC

ADCIRC is one of the most validated and widely used ocean circulation models [13]. It is developed and maintained by a large community of users and developers. Implemented with Message Passing Interface (MPI), ADCIRC has great parallel scalability in order to facilitate rapid computation of large, complex problems [88]. ADCIRC solves the Generalized Wave Continuity Equations (GWCE) [89, 49] numerically. It discretizes the equations in space using the finite element method and in time using the finite difference method. ADCIRC can be run either as a two-dimensional depth integrated (2DDI) model or as a three-dimensional model. In this study, a 2DDI model on an unstructured triangular mesh was used to illustrate the RNN runoff/ADCIRC coupled simulation.

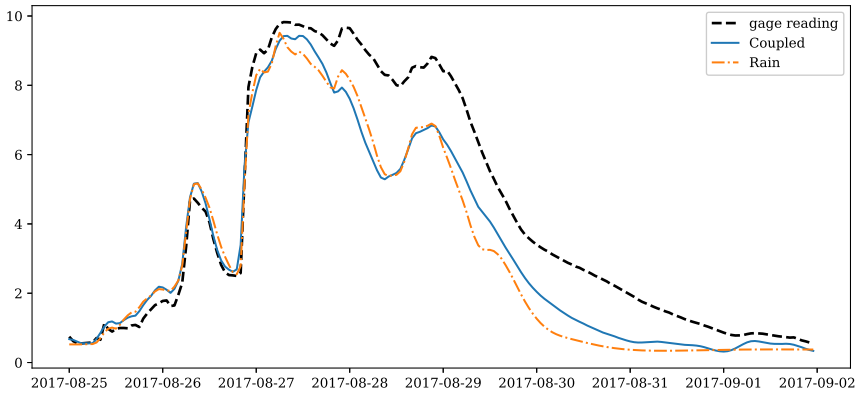


Figure 4.5: Prediction of river stage at tidal gage during Hurricane Harvey.

4.3.2 Coupling framework

Once trained, the coupled RNN model was dynamically coupled with ADCIRC. See Figure 4.6 for the diagram of our proposed coupling scheme. During a coupled simulation, the information exchange is realized by dynamically updating the flux/elevation boundary condition of ADCIRC at the river outlet and water level input of the RNN model. The information exchange is triggered every specified time interval. The exchange scheme can either start from ADCIRC or RNN, and both the models may use different time step sizes. For illustration, if RNN drives ADCIRC first (i.e., 'A' and 'B' are respectively the RNN and ADCIRC in the figure), then the RNN modifies the boundary conditions of ADCIRC for its current coupled time interval, whereas ADCIRC modifies the boundary conditions of the RNN for its next coupled interval. Since the RNN model has one hour time step which is longer than the common time step of ADCIRC, the exchange boundary conditions were exchanged every hour.

The RNN-ADCIRC coupling framework was implemented in Python and

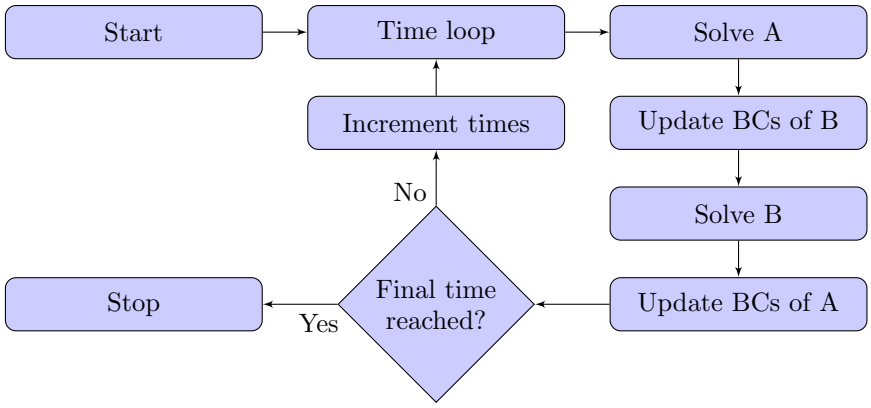


Figure 4.6: Two-way coupling scheme of RNN runoff model and ADCIRC. A and B above may be either ADCIRC or RNN.

open sourced[1]. The storm surge part of the model requires the Python interface of ADCIRC, i.e., pyADCIRC [58]. Also, the deep learning rainfall runoff part of the model requires PyTorch.

4.3.3 Numerical Experiment

To test the RNN/ADCIRC coupled model, and to avoid long computational time of the ADCIRC model, a coarse grid was constructed for Galveston Bay that connects Brays Bayou where the flux boundary condition is imposed. Hurricane Harvey was used as an illustrative example: the recorded water level at NOAA station #8771341, Galveston Bay Entrance, North Jetty TX was imposed at the entrance of Galveston Bay as tidal forcing.

Model verification and validation are crucial procedures for developing computational models. The developed coupled model consists of a deep neural network and a well verified and validated storm surge model. Thus, the only component left to verify is the coupling framework. Moreover, since the developed ADCIRC mesh and other inputs did not reflect the ground truth, it is not reasonable to compare the

[1]The code of can be accessed at https://github.com/UT-CHG/adcirc_nn.

predicted storm surge from the coupled simulation with realistic measurements for model validation. Two additional ADCIRC simulations were run to verify and validate the coupling framework. Both simulations used the same input as the ADCIRC part of the dynamically coupled simulation, except the flux boundary condition. In the first simulation, referred to as verification run, the boundary condition was set to the predicted hydrograph by the RNN/ADCIRC coupled model. In the second simulation, referred to as validation run, the flux boundary condition was set according to the USGS measurements. The verification run was used to verify that the coupling framework is correctly implemented. Under this circumstance, the verification run should provide the exact same input to ADCIRC as during the coupled simulation. And the validation run was used to validate that the coupling framework provides accurate prediction under our simplification and assumption. Simulation results are compared at the validations points as shown in Figure 4.1 for verification and validation.

4.3.4 Results

Dynamic coupling of the coupled RNN model and ADCIRC was tested with Hurricane Harvey. The dynamically coupled river discharge prediction was shown in Figure 4.3a and 4.3b together with the one-way coupled prediction. The discrepancy between the dynamically coupled prediction and the one-way RNN coupled prediction using the NOAA record is very small. Zoom-in views (see Figure 4.3b) show the period where the dynamically coupled prediction is slightly less accurate than the prediction using measurements as input. As a matter of fact, during the coupled simulation, the predicted water level rise at Morgans Point by ADCIRC is more than 50% lower than the NOAA reading, since the ADCIRC model is simplified and did not account for the wind forcing and riverine input other than Brays Bayou. However, the prediction from RNN runoff model side is still relatively ac-

curate thanks to (1) the effect of storm surge from Galveston Bay on runoff at the fresh gage is minor compared to the rainfall (2) storm surge is not as severe in Galveston Bay compared to other affected areas during Hurricane Harvey. The similarity between the two predictions does not indicate that the trained RNN model is insensitive to downstream water level. To show this, a hypothetical storm, composed of precipitation from an event started Dec. 1, 2017 and downstream water level profile during Hurricane Harvey (see Figure 4.7), was created. The predicted peak streamflow of the hypothetical event was 4.2% higher than the prediction of the original event.

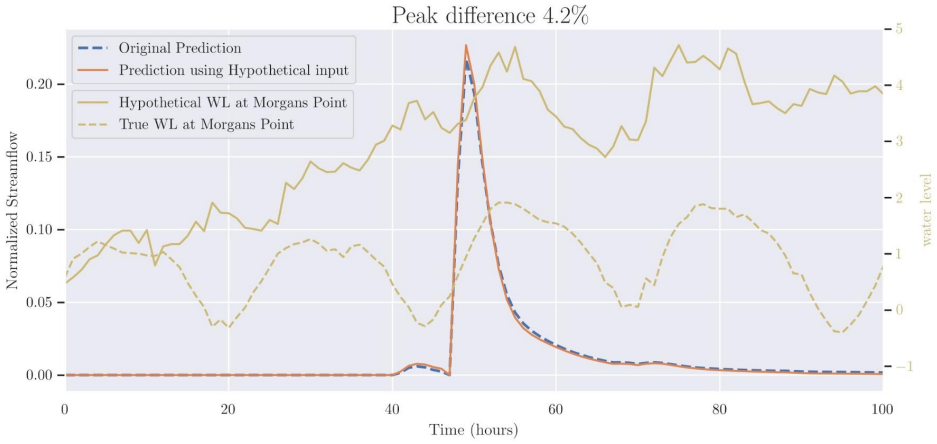


Figure 4.7: Predicted hydrograph of hypothetical storm by coupled RNN model.

To verify the implementation of the coupling framework, the ADCIRC water surface elevation predictions of the verification run were compared with the dynamically coupled run. The simulation results were exactly the same with maximum error lower than 1E-7, suggesting the implementation was correct. Furthermore, the validation run generated a WSE prediction that is very close to the coupled run. As shown in Figure 4.8, at tidal gage and Lynchburg landing (LB), the correlation between the two predictions are 0.99668 and 0.99898. At further away validation

points, the correlation between the two were even closer to 1. The trend is also consistent with our hydrologic understanding: as the location of validation points go further into the Galveston Bay, the effect of river discharge on WSE decays.

4.4 Discussion

For coastal regions, intense rainfall runoff and high storm surge are not mutually exclusive. To improve the prediction accuracy, it is necessary to combine both mechanisms in hydrologic simulation. The hybrid deep learning-computational model introduced in this chapter demonstrates the ability of an RNN to transfer precipitation as well as nonlocal downstream tidal information directly to the streamflow (or river stage) of watershed runoff. Adding the downstream tidal information to predict fresh gage discharge improved the generalization performance of the RNN model. One the other hand, adding the same downstream information can significantly improve the prediction accuracy of the tidal gage stage. Consistent results have shown us it effectively reduced the prediction bias of the RNN model in the medium stage region.

Using the proposed network architecture, the RNN runoff model can easily be coupled with ocean circulation models to simulate compound flooding events. During a coupled simulation, the RNN predicted streamflow (or stream height) can be used as the boundary condition of a computational storm surge model such as ADCIRC. On the other hand, the required downstream tidal information is supplied by the storm surge model. As an illustrative example, such a framework was developed to couple the RNN runoff model with ADCIRC and a simplified simulation of Hurricane Harvey with conducted with the developed coupler. In reality, prediction performance of the coupled model can be improved by (1) training runoff models for all other watersheds that runoff to the ocean and adding the trained models to the coupler (2) using a finer mesh for the ocean circulation model and matching the

boundary elements with the size of river.

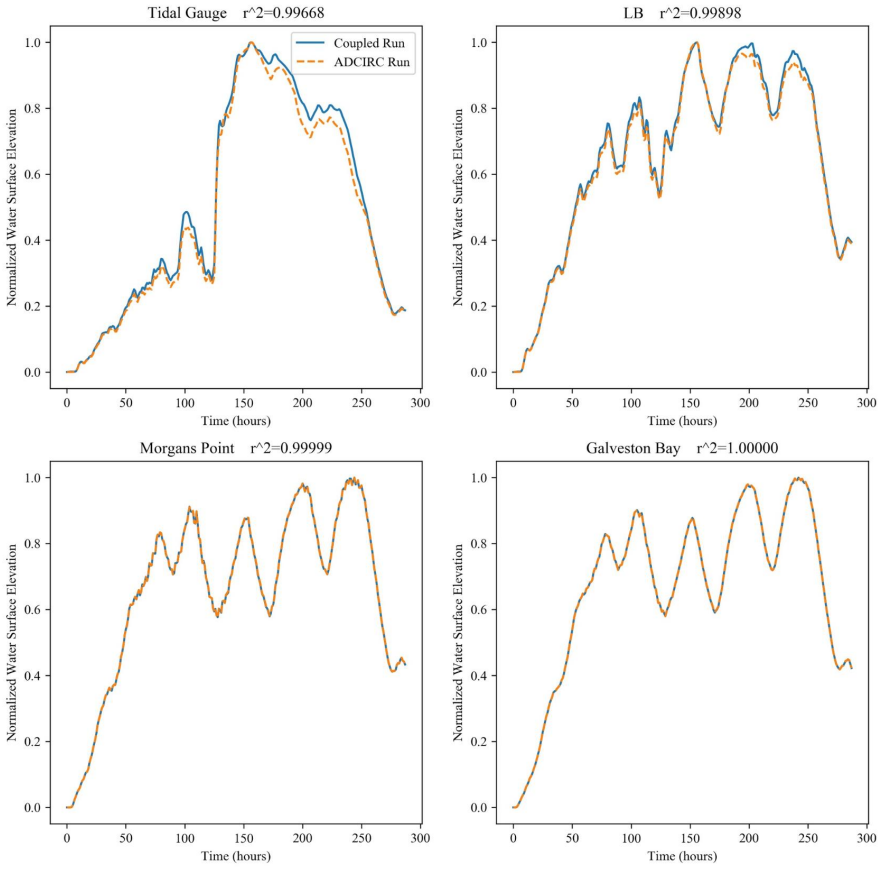


Figure 4.8: Coupler validation result. Even at the tidal gage location, the result of coupled run is close to the one from ADCIRC run with ground truth boundary condition. As we go from the upper left to the bottom right (further into Galveston Bay), the two predicted WSE curves become identical.

Chapter 5

Conclusion

In this work, given hydrologic inputs, a deep RNN was used to infer the hydrograph of rainfall runoff events. A comparison study on network architectures showed that the synced sequence input and output architecture outperforms the sequence input single output one in terms of prediction accuracy and computational efficiency. Later, downstream water level interaction with the watershed runoff was integrated into the developed RNN hydrologic model. By introducing the downstream information to the RNN hydrologic model, substantial improvement in inference accuracy was observed. Finally, through an example, the dynamic coupling of the RNN hydrologic model and an ocean circulation model (ADCIRC), was tested, verified, and validated for compound flood events.

In this study, the default implementation of LSTM network in PyTorch was used. In the future, more advanced variants of RNN architectures, e.g. Fourier recurrent units [90] and regularized piecewise linear RNNs [91], can be applied to hydrologic modeling to handle the long-term dependency within sequences and avoid complicated mathematical structure such as the LSTM network. Another direction of future study is to use data-driven or surrogate models to accelerate storm surge simulation. The generalized wave-continuity equations can be solved using physics-

informed neural networks [92]. Or the discretized governing equations can be solved by a graph neural network [93]. The trade of between the inference speed improvement and the addition computational expenses of training neural networks will be investigated in the future.

Bibliography

[1] Wei Li, Amin Kiaghadi, and Clint Dawson. High temporal resolution rainfall–runoff modeling using long-short-term-memory (lstm) networks. *Neural Computing and Applications*, pages 1–18, 2020.

[2] Wei Li, Amin Kiaghadi, and Clint Dawson. Exploring the best sequence lstm modeling architecture for flood prediction. *Neural Computing and Applications*, pages 1–10, 2020.

[3] Emanuele Bevacqua, Douglas Maraun, Michalis Ioannis Vousdoukas, E Voukouvalas, M Vrac, L Mentaschi, and M Widmann. Higher probability of compound flooding from precipitation and storm surge in europe under anthropogenic climate change. *Science advances*, 5(9):eaaw5531, 2019.

[4] Thomas Wahl, Shaleen Jain, Jens Bender, Steven D Meyers, and Mark E Luther. Increasing risk of compound flooding from storm surge and rainfall for major us cities. *Nature Climate Change*, 5(12):1093–1097, 2015.

[5] Avantika Gori, Ning Lin, and James Smith. Assessing compound flooding from landfalling tropical cyclones on the north carolina coast. *Water Resources Research*, 56(4):e2019WR026788, 2020.

[6] ZD Tessler, Charles J Vörösmarty, M Grossberg, I Gladkova, H Aizenman,

JPM Syvitski, and E Foufoula-Georgiou. Profiling risk and sustainability in coastal deltas of the world. *Science*, 349(6248):638–643, 2015.

[7] NOAA. What percentage of the american population lives near the coast. https://oceanservice.noaa.gov/facts/population.html. Accessed: 2021-04-08.

[8] National Weather Service. Hurricane florence: September 14, 2018. https://www.weather.gov/ilm/HurricaneFlorence. Accessed: 2021-04-08.

[9] Félix L Santiago-Collazo, Matthew V Bilskie, and Scott C Hagen. A comprehensive review of compound inundation models in low-gradient coastal watersheds. *Environmental Modelling & Software*, 119:166–181, 2019.

[10] Mark Loveland, Clint N Dawson, Hanadi S Rifai, Shubhra Misra, Helena Mosser, Alessandro Parola, et al. Developing a modeling framework to simulate compound flooding: When storm surge interacts with riverine flow. *Frontiers in Climate*, 2:35, 2020.

[11] Gajanan Choudhary. *Coupled atmospheric, hydrodynamic, and hydrologic models for simulation of complex phenomena*. PhD thesis, University of Texas, 2019.

[12] Charles W Downer and Fred L Ogden. Gssha: Model to simulate diverse stream flow producing processes. *Journal of Hydrologic Engineering*, 9(3):161–174, 2004.

[13] Jason G Fleming, Crystal W Fulcher, Richard A Luettich, Brett D Estrade, Gabrielle D Allen, and Harley S Winer. A real time storm surge forecasting system using adcirc. In *Estuarine and Coastal Modeling (2007)*, pages 893–912. 2008.

[14] FE Hicks and T Peacock. Suitability of hec-ras for flood forecasting. *Canadian water resources journal*, 30(2):159–174, 2005.

[15] Streamline Technologies Inc. Channel interconnected and model pond routing version 3.10. http://www.streamnologies.com/products/icpr/icpr.htm. Accessed: 2021-04-08.

[16] US Army Corps of Engineers Engineer Research and Development Center. The adaptive hydraulics (adh) modeling system. https://www.erdc.usace.army.mil/Locations/CHL/AdH/. Accessed: 2021-04-08.

[17] Walter F Silva-Araya, Félix L Santiago-Collazo, Juan Gonzalez-Lopez, and Javier Maldonado-Maldonado. Dynamic modeling of surface runoff and storm surge during hurricane and tropical storm events. *Hydrology*, 5(1):13, 2018.

[18] MV Bilskie and SC Hagen. Defining flood zone transitions in low-gradient coastal regions. *Geophysical Research Letters*, 45(6):2761–2770, 2018.

[19] Han Song Tang, I Steven, Jy Chien, Marouane Temimi, Cheryl Ann Blain, Qu Ke, Liuhui Zhao, and Simon Kraatz. Vulnerability of population and transportation infrastructure at the east bank of delaware bay due to coastal flooding in sea-level rise conditions. *Natural hazards*, 69(1):141–163, 2013.

[20] Xiaohui Yuan, Chen Chen, Xiaohui Lei, Yanbin Yuan, and Rana Muhammad Adnan. Monthly runoff forecasting based on lstm–alo model. *Stochastic Environmental Research and Risk Assessment*, 32(8):2199–2212, 2018.

[21] Haishen Lü, Ting Hou, Robert Horton, Yonghua Zhu, Xi Chen, Yangwen Jia, Wen Wang, and Xiaolei Fu. The streamflow estimation using the xinanjiang rainfall runoff model and dual state-parameter estimation method. *Journal of Hydrology*, 480:102–114, 2013.

[22] Guangyuan Kan, Cheng Yao, Qiaoling Li, Zhijia Li, Zhongbo Yu, Zhiyu Liu, Liuqian Ding, Xiaoyan He, and Ke Liang. Improving event-based rainfall-runoff

simulation using an ensemble artificial neural network based hybrid data-driven model. *Stochastic environmental research and risk assessment*, 29(5):1345–1370, 2015.

[23] Saadat Ayub Khan and Linda See. Rainfall-runoff modelling using data driven and statistical methods. In *2006 international conference on advances in space technologies*, pages 16–20. IEEE, 2006.

[24] Francesco Granata, Rudy Gargano, and Giovanni De Marinis. Support vector regression for rainfall-runoff modeling in urban drainage: A comparison with the epa's storm water management model. *Water*, 8(3):69, 2016.

[25] Kuo-lin Hsu, Hoshin Vijai Gupta, and Soroosh Sorooshian. Artificial neural network modeling of the rainfall-runoff process. *Water resources research*, 31(10):2517–2530, 1995.

[26] Christian W Dawson and Robert Wilby. An artificial neural network approach to rainfall-runoff modelling. *Hydrological Sciences Journal*, 43(1):47–66, 1998.

[27] KP Sudheer, AK Gosain, and KS Ramasastri. A data-driven algorithm for constructing artificial neural network rainfall-runoff models. *Hydrological processes*, 16(6):1325–1330, 2002.

[28] P Hettiarachchi, MJ Hall, and AW Minns. The extrapolation of artificial neural networks for the modelling of rainfall—runoff relationships. *Journal of Hydroinformatics*, 7(4):291–296, 2005.

[29] Sanaga Srinivasulu and Ashu Jain. A comparative analysis of training methods for artificial neural network rainfall–runoff models. *Applied Soft Computing*, 6(3):295–306, 2006.

[30] Chih-Chieh Young and Wen-Cheng Liu. Prediction and modelling of rainfall–runoff during typhoon events using a physically-based and artificial neural network hybrid model. *Hydrological Sciences Journal*, 60(12):2102–2116, 2015.

[31] Marcel van Gerven and Sander Bohte. *Artificial neural networks as models of neural information processing*. Frontiers Media SA, 2018.

[32] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929, 2013.

[33] Olivier Nerrand, Pierre Roussel-Ragot, Léon Personnaz, Gérard Dreyfus, and Sylvie Marcos. Neural networks and nonlinear adaptive filtering: unifying concepts and new algorithms. *Neural computation*, 5(2):165–199, 1993.

[34] V Taver, A Johannet, V Borrell-Estupina, and S Pistre. Feed-forward vs recurrent neural network models for non-stationarity modelling using data assimilation and adaptivity. *Hydrological Sciences Journal*, 60(7-8):1242–1265, 2015.

[35] Anne Johannet, Valérie Borrell Estupina, Séverin Pistre, et al. Neural networks for karst groundwater management: case of the lez spring (southern france). *Environmental Earth Sciences*, 74(12), 2015.

[36] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[37] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[38] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *9th International Conference on Artificial Neural Networks*, pages 850–855, 1999.

[39] Jürgen Schmidhuber and Sepp Hochreiter. Long short-term memory. *Neural Comput*, 9(8):1735–1780, 1997.

[40] Zakaria Mhammedi, Andrew Hellicar, Ashfaqur Rahman, Kasirat Kasfi, and Philip Smethurst. Recurrent neural networks for one day ahead prediction of stream flow. In *Proceedings of the Workshop on Time Series Analytics and Applications*, pages 25–31, 2016.

[41] Frederik Kratzert, Daniel Klotz, Claire Brenner, Karsten Schulz, and Mathew Herrnegger. Rainfall–runoff modelling using long short-term memory (lstm) networks. *Hydrology and Earth System Sciences*, 22(11):6005–6022, 2018.

[42] Zhaoyang Liu, Weigang Xu, Jun Feng, Shivakumara Palaiahnakote, Tong Lu, et al. Context-aware attention lstm network for flood prediction. In *2018 24th international conference on pattern recognition (ICPR)*, pages 1301–1306. IEEE, 2018.

[43] Xuan-Hien Le, Hung Viet Ho, Giha Lee, and Sungho Jung. Application of long short-term memory (lstm) neural network for flood forecasting. *Water*, 11(7):1387, 2019.

[44] Frederik Kratzert, Daniel Klotz, Guy Shalev, Günter Klambauer, Sepp Hochreiter, and Grey Nearing. Benchmarking a catchment-aware long short-term memory network (lstm) for large-scale hydrological modeling. *Hydrology and Earth System Sciences Discussions*, pages 1–32, 2019.

[45] Caihong Hu, Qiang Wu, Hui Li, Shengqi Jian, Nan Li, and Zhengzheng Lou.

Deep learning with a long short-term memory networks approach for rainfall-runoff simulation. *Water*, 10(11):1543, 2018.

[46] I-Feng Kao, Yanlai Zhou, Li-Chiu Chang, and Fi-John Chang. Exploring a long short-term memory based encoder-decoder framework for multi-step-ahead flood forecasting. *Journal of Hydrology*, 583:124631, 2020.

[47] Indrastanti R Widiasari, Lukito Edi Nugoho, Rissal Efendi, et al. Context-based hydrology time series data for a flood prediction model using lstm. In *2018 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, pages 385–390. IEEE, 2018.

[48] Mohammad Karamouz, Ali Razmi, Sara Nazif, and Zahra Zahmatkesh. Integration of inland and coastal storms for flood hazard assessment using a distributed hydrologic model. *Environmental Earth Sciences*, 76(11):1–17, 2017.

[49] J Casey Dietrich, Seizo Tanaka, Joannes J Westerink, CN Dawson, RA Luettich, Marcel Zijlema, Leo H Holthuijsen, JM Smith, LG Westerink, and HJ Westerink. Performance of the unstructured-mesh, swan+ adcirc model in computing hurricane waves and surge. *Journal of Scientific Computing*, 52(2):468–497, 2012.

[50] Justin Joyce, Ni-Bin Chang, Rahim Harji, Thomas Ruppert, and Peter Singhofen. Cascade impact of hurricane movement, storm tidal surge, sea level rise and precipitation variability on flood assessment in a coastal urban watershed. *Climate Dynamics*, 51(1):383–409, 2018.

[51] Li H Erikson, Andrea C O'Neill, and Patrick L Barnard. Estimating fluvial discharges coincident with 21st century coastal storms modeled with cosmos. *Journal of Coastal Research*, 85:791–795, 2018.

[52] Liv Herdman, Li Erikson, and Patrick Barnard. Storm surge propagation and flooding in small tidal rivers during events of mixed coastal and fluvial influence. *Journal of Marine Science and Engineering*, 6(4):158, 2018.

[53] Brian Blanton, Kendra Dresback, Brian Colle, Randy Kolar, Humberto Vergara, Yang Hong, Nicholas Leonardo, Rachel Davidson, Linda Nozick, and Tricia Wachtendorf. An integrated scenario ensemble-based framework for hurricane evacuation modeling: Part 2—hazard modeling. *Risk analysis*, 40(1):117–133, 2020.

[54] Hwai-Ping Cheng, Jing-Ru Cheng, Robert Hunter, and Hsin-Chi Lin. Demonstration of a coupled watershed-nearshore model. 2010.

[55] Benjamin Bass and Philip Bedient. Surrogate modeling of joint flood risk across coastal watersheds. *Journal of Hydrology*, 558:159–173, 2018.

[56] Sang-Jin Park and Dong-Kun Lee. Prediction of coastal flooding risk under climate change impacts in south korea using machine learning algorithms. *Environmental Research Letters*, 15(9):094052, 2020.

[57] Jon French, Robert Mawdsley, Taku Fujiyama, and Kamal Achuthan. Combining machine learning with computational hydrodynamics for prediction of tidal surge inundation at estuarine ports. *Procedia IUTAM*, 25:28–35, 2017.

[58] Gajanan K Choudhary and Clint Dawson. pyadcirc: A python interface for accessing functions and variables of adcirc in python. 2020.

[59] Brays bayou. `https://www.hcfcd.org/projects-studies/brays-bayou/`, 2019. Accessed: 2019-3-7.

[60] NOAA National Centers for Environmental Information (NCEI). U.S. billion-dollar weather and climate disasters. `https://www.ncdc.noaa.gov/billions/`, 2018. Accessed: 2018-05-23.

[61] Harris county flood warning system. `https://www.harriscountyfws.org/`, 2019. Accessed: 2019-3-7.

[62] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[63] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[64] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[65] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[66] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[67] Ilya Sutskever. *Coupled atmospheric, hydrodynamic, and hydrologic models for simulation of complex phenomena*. PhD thesis, University of Toronto Toronto, Canada, 2013.

[68] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR, 2013.

[69] Jason Brownlee. A gentle introduction to the rectified linear unit (relu). *Machine learning mastery*, 6, 2019.

[70] Olivier Bousquet. The tradeoffs of large scale learning. In Suvrit Sra, Sebastian Nowozin, and Stephen J Wright, editors, *Optimization for machine learning*, pages 351–368. Mit Press, 2012.

[71] Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes. In *International Conference on Machine Learning*, pages 6677–6686. PMLR, 2019.

[72] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[73] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[74] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

[75] Peter Bühlmann and Sara Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

[76] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[77] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[78] Charles W Downer and Fred L Ogden. Gridded surface subsurface hydrologic analysis (gssha) user's manual; version 1.43 for watershed modeling system 6.1. Technical report, Engineer Research and Development Center, Vicksburg, MS, Coastal and Hydraulics Laboratory, 2006.

[79] Deva K Borah, Ebrahim Ahmadisharaf, G Padmanabhan, Sanaz Imen, and Yusuf M Mohamoud. *Watershed models for development and implementation of total maximum daily loads*. PhD thesis, American Society of Civil Engineers, 2018.

[80] Timothy Fry and Reed Maxwell. Using a distributed hydrologic model to improve the green infrastructure parameterization used in a lumped model. *Water*, 10(12):1756, 2018.

[81] Chad Furl, Dawit Ghebreyesus, and Hatim Sharif. Assessment of the performance of satellite-based precipitation products for flood events across diverse spatial scales using gssha modeling system. *Geosciences*, 8(6):191, 2018.

[82] Edsel B Daniel, Janey V Camp, Eugene J LeBoeuf, Jessica R Penrod, Mark D Abkowitz, and James P Dobbins. Watershed modeling using gis technology: A critical review. *Journal of Spatial Hydrology*, 10(2), 2011.

[83] Göran Lindström, Barbro Johansson, Magnus Persson, Marie Gardelin, and Sten Bergström. Development and test of the distributed hbv-96 hydrological model. *Journal of hydrology*, 201(1-4):272–288, 1997.

[84] François Anctil, Nicolas Lauzon, Vazken Andréassian, Ludovic Oudin, and Charles Perrin. Improvement of rainfall-runoff forecasts through mean areal rainfall optimization. *Journal of hydrology*, 328(3-4):717–725, 2006.

[85] C Fanelli, P Fanelli, and L Fenstermacher. Noaa water level and meteorological data report—hurricane ike. *US Department of Commerce, National*

Oceanic and Atmospheric Administration, National Ocean Service Center for Operational Oceanographic Products and Services, 2009.

[86] Maria Grazia De Giorgi, Stefano Campilongo, Antonio Ficarella, and Paolo Maria Congedo. Comparison between wind power prediction models based on wavelet decomposition with least-squares support vector machine (ls-svm) and artificial neural network (ann). *Energies*, 7(8):5251–5272, 2014.

[87] Vito Iacobellis. Probabilistic model for the estimation of t year flow duration curves. *Water Resources Research*, 44(2), 2008.

[88] JC Dietrich, M Zijlema, JJ Westerink, LH Holthuijsen, C Dawson, RA Luettich Jr, RE Jensen, JM Smith, GS Stelling, and GW Stone. Modeling hurricane waves and storm surge using integrally-coupled, scalable computations. *Coastal Engineering*, 58(1):45–65, 2011.

[89] Richard Albert Luettich and Joannes J Westerink. *Formulation and numerical implementation of the 2D/3D ADCIRC finite element model version 44. XX*. Citeseer, 2004.

[90] Jiong Zhang, Yibo Lin, Zhao Song, and Inderjit Dhillon. Learning long term dependencies via fourier recurrent units. In *International Conference on Machine Learning*, pages 5815–5823. PMLR, 2018.

[91] Dominik Schmidt, Georgia Koppe, Zahra Monfared, Max Beutelspacher, and Daniel Durstewitz. Identifying nonlinear dynamical systems with multiple time scales and long-range dependencies. *arXiv preprint arXiv:1910.03471*, 2019.

[92] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[93] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.

Vita

Wei Li was born in Urumqi, Xinjiang, China. He received the Bachelor of Science degree in Aeronautical and Astronautical Engineering from Shanghai Jiao Tong University in 2014 and the Master of Science in Aerospace Engineering from University of Illinois at Urbana-Champaign in 2015. He started the PhD program in the Computation Science, Engineering, and Mathematics at the University of Texas at Austin in August 2016.

Permanent Address: westli5678@gmail.com

This dissertation was typeset with LaTeX 2_ε[1] by the author.

[1] LaTeX 2_ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this dissertation were written by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin, and extended by Bert Kay, James A. Bednar, and Ayman El-Khashab.